7-28-1987

Computer Aided Instruction and a Graphical Authoring System

Catherine Lofgren Goodykoontz

Rochester Institute of Technology
School of Computer Science and Technology

Computer Aided Instruction
and a
Graphical Authoring System

by
Catherine Lofgren Goodykoontz

A thesis, submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements of the degree of
Master of Science in Computer Science

Approved by:

Name Illegible 7/28/87

Chris Comte 7/28/87

Peter H. Lutz 7/28/87

Authorization to Reproduce

Computer Aided Instruction and a Graphical Authoring System

Permission is hereby granted to the Wallace Memorial Library, of the Rochester Institute of Technology, to reproduce my thesis, in whole or in part, provided that the reproduction will not be for commercial use or profit. Written notification shall be provided for each such request at the address below.

Signed: Catherine L. Goodykoontz

Date: July 28, 1987

Catherine L. Goodykoontz

Abstract

In the early 1960's, the computer was first used as a means of delivering instruction to college students. Today, the computer as an educational tool has been integrated into curricula from preschool to the university. This thesis focuses upon the specification and implementation of a Graphical Authoring System which is used to create, deliver and administrate instruction of a pictorial nature (combining graphics and text). The system explores a question format different from the traditional formats of multiple choice, true-false, and fill-in-the-blank which are available in current authoring systems. The question format to be explored allows the student to identify an answer by choosing a point on the screen with a coordinate selecting device such as a mouse.

Table of Contents

1. COMPUTER AIDED INSTRUCTION

Having overcome a majority of the obstacles that hindered the use of computers in the classroom, educational software is now available in many forms and for all ages. Tools, such as authoring languages and authoring systems, aid the development of 'lessons' taught by a computer.

This chapter presents background information about computers in education, along with the preliminary objectives of the Graphical Authoring System to be developed. It is followed by chapters which give an overview of the system, discuss implementation issues and explain examples. The original specifications of data stores and processing are found in the appendix.

Courseware Taxonomy

Computer Aided Instruction (CAI) and Computer Based Education (CBE) are the two popular terms used to identify the use of the computer to instruct and educate. Courseware is the name given to educational software, its supporting documentation and sometimes its associated hardware. Courseware is categorized in [BORK81,DIGI83,GRAD83,LEKA71,ONEI81] by the method in which the subject matter is presented and/or taught.

Drill and Practice methodology tests acquired knowledge by asking questions and evaluating answers. The usual format of the questions is multiple choice, true/false or exact answer, as these are the easiest to implement. The majority of existing courseware is of this type.

The tutorial method introduces new information or is used for remediation. It is frequently occurs in conjunction with drill and practice. The tutorial method can range from a simple linear presentation of a textbook on a CRT screen to a sophisticated "programmed instruction" approach in which the student's performance on questions determines the path of further instruction or remediation. This allows the student to move quickly through information he already knows or is acquiring quickly while continually evaluating whether material is being understood before continuing to more difficult material.

Drill and practice is based on the Skinnerian learning model which uses repetitive reinforcement and habitual response as a learning method. In drill and practice and tutorial presentations, the student primarily assumes a passive role in the learning process.

The dialog method attempts to make the student a more active participant in the learning process by providing the student with primary control. The student "talks" to the computer in "English" and the computer provides responses to the student in "English". This is an attempt to simulate the Socratic method of learning, where a student is directed to correct conclusions by his tutor (the computer) through further questioning. It is difficult to provide this method in courseware for two reasons. First, it is difficult to program a computer to understand student responses in an unambiguous way. Second, the author of the courseware is limited in his ability to predict student responses. Dialog courseware allows the student to respond in limited English. It attempts to evaluate the student's incorrect responses and direct the instruction dependent on his or her misunderstandings. A default path of instruction is taken when the response is unexpected. Sometimes dialog is considered to be a tutorial method and the previously defined method is referred to as the simple tutorial.

Simulation is a type of courseware that is used to replicate real life phenomena and has a wide variety of applications. Safety and cost considerations often justify the use of simulations. Cockpit simulations are valuable in training potential pilots without taking the actual risks. Simulations of laboratory experiments can reduce the need for costly laboratory equipment and provide a safe environment for the student while still providing the student with knowledge of the outcome of his activities. Real world conditions simulated with hypothetical events can help students study social, political and economic systems.

Problem solving tools such as statistical packages, electronic spreadsheets and word processors that help the student in investigating and solving problems, are often considered to be forms of Computer Aided Instruction. In a broad sense, programming languages themselves can also be considered to be Computer Aided Instruction.

Courseware often contains some of the elements of computer managed instruction (CMI) or is under the control of a CMI system. Instruction management can vary from keeping performance scores for students for a course delivered on a microcomputer to a mainframe system utilized by an entire school which controls the student's courseware usage, monitors performance, provides resource allocation and scheduling, and does reporting to support these functions.

The Learning Process

The General Learning Model presented by Rockart and Morton [ROCK81] represents the learning cycle. The process begins with exposure to new knowledge; then the "practicing" and study of the newly taught information; then the integration with past knowledge to create generalizations and understanding that surpass memorization; and finally, the enrichment of knowledge and understanding by testing the concepts and generalizations. Rockart and Morton are aware that even at the university level the final step of the cycle often is not experienced.

A review of the pedagogical techniques presented in the following table from [ROCK81] points out that the majority of the educational experiences support the first phases of the learning cycle. The lecture, textbook, written assignment and their new counterparts all support primarily these first two phases. The enrichment phases of integration and testing are experienced infrequently through traditional as well as through the newer computer aided instruction methodologies. Techniques that dominated traditionally currently are dominating computer courseware.

Pedigogical techniques	Dominant traditional method	Newer noncomputer methods	Computer-based methods
Student "watches, listens" and learns	Lecture	Radio, charts, television, slides, magnetic tape, cassettes, films, videotape	Computer-based programmed instruction
Student "reads material" and learns	Textbooks	Paper programmed instruction, self-study modules	Computer-based programmed instruction
Students take active oral participation in learning	Case studies, class discussions (including seminars)	'voting machines'	
Students performs "written self-testing" and receives feedback	Written assignments	"voting machines", paper programmed instruction	Drill and practice, problem solving
Learner performs "hands-on" investigation of phenomena	Library investigation, laboratory experiments, real world experience		Simulation models, games, inquiry, problem solving

Generally,the following are perceived as advantages of Computer Aided Instruction over other methods.

Individualized instruction requiring concentration from, and interaction with, an individual.

- The repetition required by the method is not frustrating or tiring for a computer.

Immediate feedback to student responses.

Consistent presentation and responses from one day to the next and to all students.

For the most part, Computer Aided Instruction requires the teacher to continue to play a part in the student's learning. The computer can not take an exclusive role in instruction. Maintaining a variety of pedagogical techniques and methods is important to fully support the learning process.

The video disk is expected to have a great impact on education. When combined with a computer, it is sometimes referred to as the intelligent video disk. Since the video disk can be accessed randomly, a computer can utilize it as a database and quickly display portions of it (from one frame to a sequence of frames) and present it in a programmed instruction fashion.

Obstacles to CAI

Programmed instruction was first implemented on computers in the early 1960's. In the next two decades CAI did not have the influence on education that was expected. A combination of factors has affected the slow entry of computers in education as explained in [BORK81,ONEI81].

CAI was much more expensive than conventional methods. The instructional benefits were not great enough to justify the expense of hardware, software and course development. Even with the development of course authoring languages, such as IBM's Coursewriter and DEC's DECAL, which helped to shorten the course development time, conversion of course material to courseware was expensive.

The number of knowledgeable author sources have been limited. Subject matter experts are not interested in becoming software programmers. The subject matter expert should only need a general knowledge of the computer's capabilities as an educational tool. (The subject matter expert doesn't have to run a printing press in order to write a text book, why should he or she program a computer?) It is apparent from the generally poor quality of courseware that the communication between the educational professionals and the computer professionals has not been effective.

The absence of effective means of distribution of courseware has had a significant impact on its usage and cost effectiveness. Even excellent courseware may be difficult to justify in terms of cost if it has only been used in the locality where it was produced.

Although these obstacles have not been overcome totally, the 1980's have seen increasing activity in overcoming them. The microcomputer is expected to have the greatest impact on CAI by making computer hardware affordable [FREN80]. Educators are learning to insist on better quality courseware and are formalizing their evaluation criteria [INST84, LATA80]. Organizations such as EPIE are providing in-depth evaluations of courseware.

In comparing courseware catalogs [CHAR82, EPIE81, EPIE83] from the early 70's to those of the 80's, courseware listed in the catalogs of the 70's identified individual authors most of whom were affiliated with colleges and universities, while in the 80's the producers and distributors of courseware include traditional publishers (e.g. Scott, Foresman & Co., Readers Digest), game manufacturers (eg. Milton Bradley), computer manufacturers, new courseware companies and educational organizations (eg. CONDUIT, MECC). CAI is no longer just a cottage industry.

However, while progress in CAI development has appeared slow, previous instructional technologies have had to surmount very similar obstacles. "Over 200 years elapsed between the invention of the printing press and the large scale use of textbooks in formal institutions of higher learning." [BORK80, p.205]

Past and Present

Two government funded projects that dominated early CAI history, PLATO and TACCIT, were begun in the 1960's [KOZM78, ONEI81, LEKA71]. Both used primarily tutorial and drill and practice as their delivery techniques and both had extensive CMI functions administrating the courseware. PLATO was developed on a mainframe to allow thousands of students to use the instructional software and to monitor their performance. It retained the conventional teacher-centered approach where the teacher had control scheduling and activity. An authoring language called TUTOR was provided to be used by curriculum developers in developing courseware. TUTOR was sophisticated and it took a long time to become efficient at developing pedagogical techniques to implement by "programming" with TUTOR.

TACCIT was developed at Brigham Young University on a minicomputer expected to serve 100 to 200 students. The courseware used primarily a rule-example-practice pattern. Much more learner

control was allowed by letting the student choose his strategy for study of the components. This student-centered approach originally had low completion rates until deadlines with penalties were added.

Since the mid 1970's CAI has started to play an increasingly more significant role in education. Today the microcomputer has made its way into high schools, elementary schools, libraries, homes and even the preschool environment for educational purposes. Students are learning to program in LOGO and BASIC. Courseware is available for many topics from a vast variety of sources. Mathematics and Language Arts account for a majority of the courseware. Currently instituted curriculum requirements for computer literacy have required the education of teachers only a step ahead of their students. [INST84]

Today's courseware is developed through three categories of tools - traditional programming languages, authoring languages and authoring systems. [LOCA85, GRAH85] Authoring languages and authoring systems identify activities that are common to courseware development and provide mechanisms to implement these quickly and with less requirements for programming expertise. With each of these methods there is a loss of the versatility of capabilities of the general purpose programming languages.

Authoring languages still require a basic understanding of programming language constructs but provide statements that simplify activities such as displaying information and questions, accepting and evaluating answers and record keeping. Authoring systems guide the author through the development by providing menus of development options and can provide extensive automatic CMI functions. While authoring languages and authoring systems attempt to make courseware development more approachable by educators allowing them to produce courseware that meets their specific needs, these tools have been marketed primarily to industry for training course development and to institutions of higher learning where the expertise and money is available to invest in on-site course development. Courseware that has already been developed continues to be preferred by primary and secondary schools.[INST84]

Three current authoring systems are THE AUTHOR PLUS from Raptor Systems,Inc; InterAct from Ashton Interactive Training Corp.; and TenCORE Assistant from Computer Teaching Corp. [ASHT86, COMP86, RAPT86]. They all run on the IBM PC family and compatibles. TenCore additionally provides a tool to convert produced lessons for use on the Apple II family of PCs.

TenCORE Assistant is a menu driven authoring system that was added to the TenCORE products to make the initial development of courseware simpler while also providing the author with

an introduction to the underlying authoring language, the TenCORE Language Authoring System. Courseware produced by TenCORE Assistant is compatible with the TenCORE Language Author System which allows more versatility in courseware production through utilizing its "English-like" commands.

All three vendors provide similar features. All provide a means of identifying sequencing and branching of frames or segments of the lesson. Each provides a means for producing the segments to include text or graphics or a combination of both. THE AUTHOR PLUS provides its graphics with 94 special characters used to create pictorial representations. TenCORE and InterAct provide graphics editors and also allow the inclusion of segment from video disks. TenCORE also provides the capability to create a customized character set.

Multiple kinds of question formats are supported. All three systems minimally support Multiple Choice, True-False, and Fill-in formats. Fill-in answer evaluation extends past exact answer checking in all cases and includes looking for keywords and checking for misspellings. Responses to answers can be specified for right answers as well as for expected wrong answers. In addition to the keyboard and CRT, a variety of Input/Output devices are supported among the systems - speech processors, mouses, light pens, and video tape.

Authors are supported by reports to aid in verification of completeness and correctness of the produced course. Students can be given the ability to control their progression through the course. CMI packages are provided to provide record keeping and statistical reporting on student performance within the limits of the systems' storage capacities.

In response to the need for a less expensive and less complex authoring system, Ashton has provided a striped-down version of InterAct called ASAP. It eliminates some of the most costly features - video device interfacing, speech processor interfacing, student control, and text and graphics overlaying.

Evaluation of Courseware

Although developments in hardware, especially the microcomputer, have had tremendous impact on making computers accessible as an instructional tool in educational institutions, it is generally agreed that courseware quality is low. As educators develop acceptance criteria for

evaluating the instructional effectiveness of the courseware, become more selective in their purchasing of courseware, and become more involved in courseware development it is expected that quality will greatly improve.

In the evaluation of courseware, the following characteristics should be considered in light of the particular curriculum needs to be fulfilled [JONE85, LATA82, LATB82, EPIE81, EPIE83].

Graphics. Graphics are most often used to enhance instruction as a motivational tool. Graphics of this type should be evaluated as to whether it retains motivation or distracts attention from the instructional content of the courseware. Animated graphics should be considered in the same manner. Graphics can also serve as an integral part of the instruction. Just as pictures and diagrams in a book, computer graphics is a valuable tool to use to communicate information in an easily understood form.

Control. Courseware should be capable of responding to individual needs immediately in a method more sophisticated than the linear textbook approach, taking advantage of the computer's capability to branch dependent on conditions. The user and/or teacher can be given capabilities such as limiting answer response time, controlling rate of display (reading rate), sequencing the topics studied, exiting the program at any time, reviewing instructions, and requesting help and hints. Current courseware usually only provides for controlling the rate of display and the ability to exit.

Feedback. Student responses should be replied to with nonthreatening and friendly feedback in a patient manner. Audible response to a wrong answer can often be threatening and audible responses of any kind can often be distracting. A variance in the feedback that the student receives may be more effective than responding in the same manner to all right answers (and all wrong answers). Feedback should not positively reinforce the wrong answer and should be suitable for the age and the level of the student.

Presentation. The screen format should be uncluttered and easy to read. The expected duration of the lesson should be suitable for the attention span of the intended student. Using random generation in the presentation of drill and practice questions makes each execution of the courseware unique and a student can not simply memorize the ordering of the questions.

Content. The subject matter presented by the courseware should be accurate, presented using correct grammar and sentence structure, and should be free of ethnic, religious and sexual bias. The ability of the teacher to alter the contents can often be useful in extending the usefulness of the courseware. For example, allowing the teacher to change the spelling list for a spelling drill.

Managed instruction. Courseware varies greatly in the amount of management capabilities provided. The program could collect and store information during the learning session that the teacher can look at later to review student performance. This saved information is sometimes in the format that allows the student to exit the session after partially completing it and to return and resume where he left off. Some courseware evaluates student performance and automatically adjusts difficulty levels; others allow the difficulty to be manually adjusted by the instructor or student.

Packaging & Documentation. All courseware should be provided with guides containing operating descriptions for both the teacher and the student. The teacher's guide should also provide direction in strategy of use and supporting instructional activities. Packaging should provide a means to keep the components of the courseware together and the components should be labelled well.

The most effective way to evaluate courseware is to have a hands-on preview of it. Many distributors allow a 30 day preview period.

Graphical Authoring System

The Graphical Authoring System to be developed as the subject of this thesis provides the common features of existing authoring systems, responds to the courseware evaluation criteria specified in the previous section, and focuses upon a question type that is not traditionally provided by authoring systems. The Graphical Authoring System provides the framework for an educator to "author" quality courseware.

The question type that is provided allows the student to choose a point on the screen to identify an answer. The selected point is then evaluated to determine if it is within an area which is considered correct or is within an area which has been identified as an anticipated wrong answer. The author's designated response is displayed to the student. This type of question encourages the production of courseware in which graphics is an integral part of the presentation.

The author can allow the student to choose the direction of the course at specific points in the course delivery, present remedial information where appropriate, identify feedback to the student's answer selection, and build a database of questions from which to randomly select questions. Although the mechanisms are provided to incorporate these elements of control, feedback and presentation, the author's educational expertise is called upon to tailor the course to the needs of the intended student audience.

Each course developed with the Graphical Authoring System will be presented to the student in a similar manner, regardless of the course content. The author of the course has the additional responsibility of documenting the educational intent of the courseware. Use of the Graphical Authoring System is described in Chapter 3 and Appendix A.

Since the author of a course often will be unaccustomed to the computer environment and not knowledgeable in programming, this system focuses upon simplicity of use and partitioning of the authoring tasks.

The following chapter described the authoring, course delivery, and course administration capabilities that make up the Graphical Authoring System. Subsequent chapters focus on the subset of the system that was implemented.

2. General Description of the GRAPHICAL AUTHORING SYSTEM

The following is a brief description of the system specified to meet the objectives outlined in the previous chapter. Detailed specifications can be found in the appendix.

The Graphical Authoring System is comprised of three major components.

* Creation of courses,
* Administration of courses, and
* Execution of courses

Creation of courses. The author of a course defines the content of each "step" in the course and the sequence of the presentation of the steps. Three types of course steps are supported - Question Steps, Lesson Steps and Option Steps. A Question Step consists of a collection of questions and the criteria that specify what is successful or unsuccessful performance in the answering of the questions. In order to reduce repeated presentation of the same question, the collection of questions that is specified can be significantly larger than the number needed to meet performance criteria. A particular question can be specified as contained in more than one collection of questions and the same collection of questions can be specified to be used by different Question Steps with different performance criteria. These two capabilities allow for the use of one question multiple times within and across courses.

When identifying the Question Step's performance criteria, the author specifies: the minimum number of questions that must be answered for the step to be completed, the number of questions within the last "so many" asked that must be answered correctly to be considered a successful completion, the maximum number of questions that are asked, and the number of attempts that are allowed on the questions. If the criteria for successful completion are not met when the maximum number of questions have been asked, then it is considered an unsuccessful completion.

A Lesson Step consists of a lesson screen. The material presented may be new or remedial, depending on how the course steps are sequenced. A particular lesson screen can be used by more than one lesson step.

The author sequences the course presentation by identifying the next step after the completion of a Lesson Step and the next steps, one for successful completion and one for unsuccessful completion, after completion of a Question Step. The Option Step provides a mechanism to allow the student to indicate a choice in the direction of the course presentation. The author designates the choices (areas on the screen display that can be selected by the student) and at which step the course presentation proceeds for each choice.

All three course step types are directed toward graphical representation and presentation. This system should be used when this type of presentation is useful. Question, Lesson and Option Screens are all constructed using a collection of graphical directives to produce line drawings or "pictures". Text is allowed and is expected to be incorporated with the pictures, but the system is not intended for creating entire screens of text.

Possible answers, both right and wrong, are specified for each question. The answers are specified by areas (collections of points within boundaries) on the question display. Responses to answer areas and default responses for points not included within the answer areas are specified. The student's response to a question is made by selecting a point on the screen.

EXAMPLE. The question is a drawing of New York State with the text "Please point to Rochester, NY". The area on the drawing east of Rochester is specified to respond with "You're too far east" The area around Long Island responds with "You've pointed to downstate New York. Rochester is in upstate New York". The area of Rochester responds with "That's correct. That's Rochester NY, nicknamed the Flower City" All remaining points on the screen are specified to respond with "Rochester is along the Genessee River close to Lake Ontario".

Option choice areas are specified in much the same way as answer areas except that instead of a response being specified for the area, the next step in the course sequence is specified.

The author can use the system to develop a variety of presentation types to meet the specific needs of the student audience. A course can be any of the following or variations:

- One Question Step with a large collection of questions to allow the student to test his understanding of material presented elsewhere (drill and practice)
  Lesson screens followed by question sessions with remedial lessons if performance on the questions is not good.
- Question Steps of one question leading to other questions dependent on whether the question was answered correctly.

Administration of courses.

The Administrator can:

specify which student has access to which course

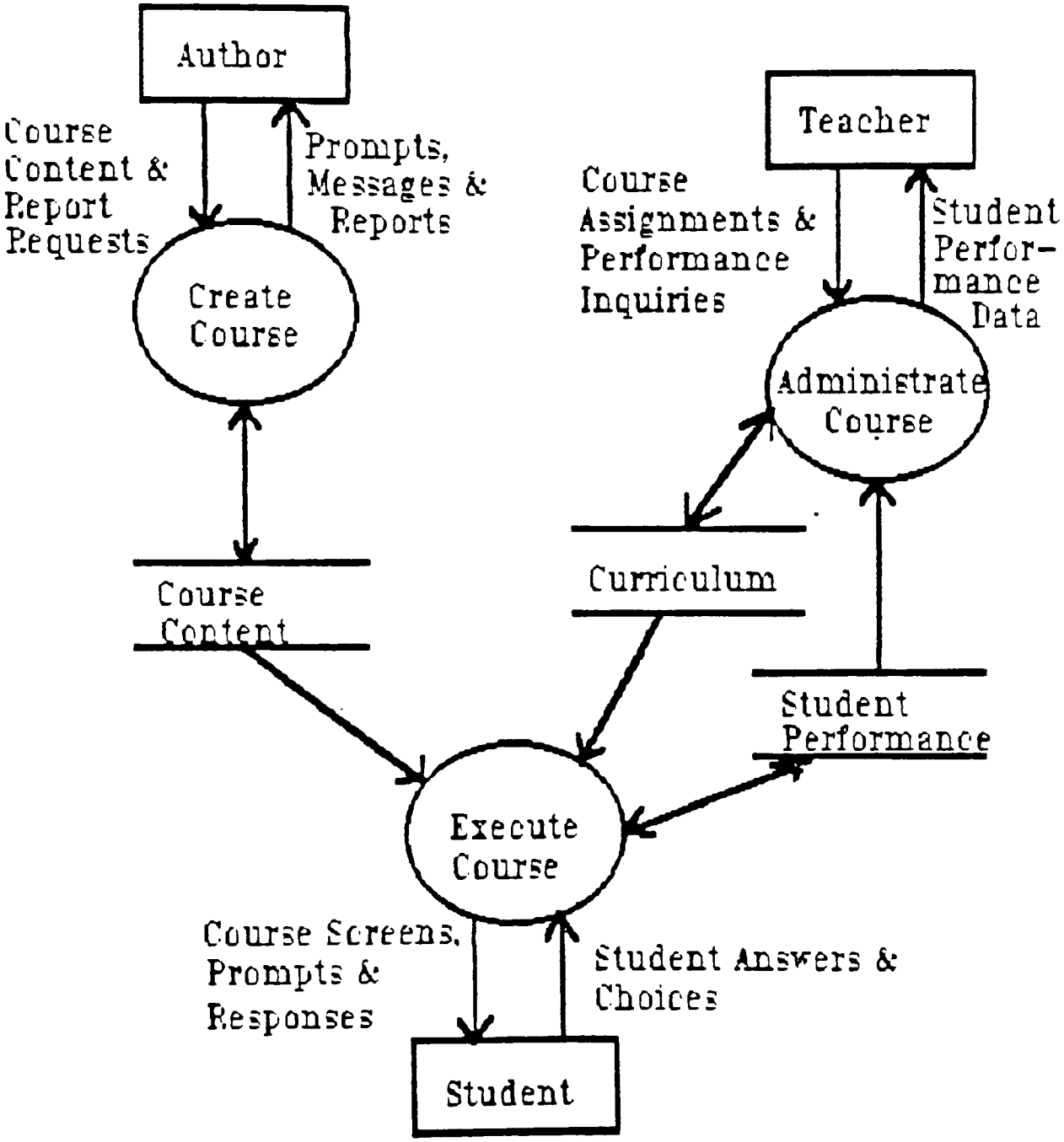
receive reports on student status

- success/unsuccessful completion of a course and when completed
- current step of course started but not completed
- courses assigned but not started

Execution of courses. Within the access limitations specified by the Administrator, a student can "take a course". The sequence of the presentation of lessons, questions and option selections follows the organization constructed during the development of the course. Performance on the course and current position in the course are updated as the student is taking the course. The student has the ability to discontinue the course at any time and start at the beginning of the current step at a later time.

Informational and directive messages are displayed in designated areas on the screen to guide the student through the coursework. After a question is answered, an informational message is displayed for the point chosen by the student. The directive message would direct the student to press a key to continue if he answered correctly, to press a key to see the correct answer or to try again at answering if he is allowed more attempts. For a lesson display, the directive message would indicate the student should press a key when he is ready to continue. The Option display directive message requests the student to select an option.

The following figure is a data flow diagram that shows the way in which the author, teacher and student interact with the Graphical Authoring System.



3. THE USER INTERFACE

This chapter focuses on describing the implemented Graphical Authoring System. It presents what was implemented and how to use the system.

Scope of Implementation

The following is a summary of the Graphical Authoring System functions from the original specification that were implemented. Please refer to Table 1-3 in Appendix C for further details.

When authoring courseware, the author can

add a new course or modify an existing one.

list the courses currently in the database.

- add, modify, display and delete the specifications for a question step in a course. (In this version 1 implementation, only one course frame can be assigned to a question step instead of a group of questions).

create, modify, display, and delete frames (screen displays).

- add, display and delete graphical answer specifications for question frames.
- add, modify, display and delete the specification of the starting step of a course.

print content and sequence reports for a course. These reports aid in the identification of the correctness and completeness of course development.

When taking a course, the student

- can exit an uncompleted course and resume at the last presented course step in the next session.

can identify the course he desires to take.

will receive correct and incorrect answer feedback as specified by the author of the course.

will be advised of completion status of successful or unsuccessful upon finishing the course.

None of the administration functions have been implemented.

USING THE GRAPHICAL AUTHORING SYSTEM

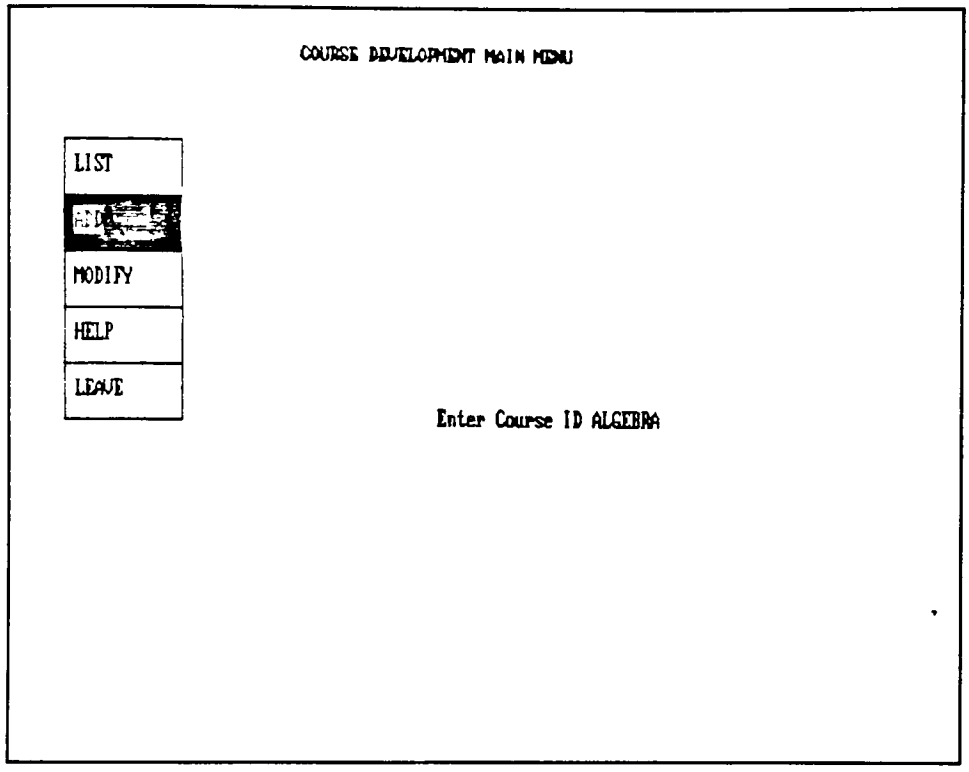
The Graphical Authoring System consists of four programs that initialize an empty course database; allow the author to develop a course; allow the author to convert a screen display created with Microsoft Paintbrush into the format of a Graphical Authoring System course frame; and allow a student to take a course. Descriptions of how to use these programs follows.

Creating the initial database (MAKEDB.COM)

Type MAKEDB in order to create the initial empty database. The created database files have the extensions of NDX and DAT.

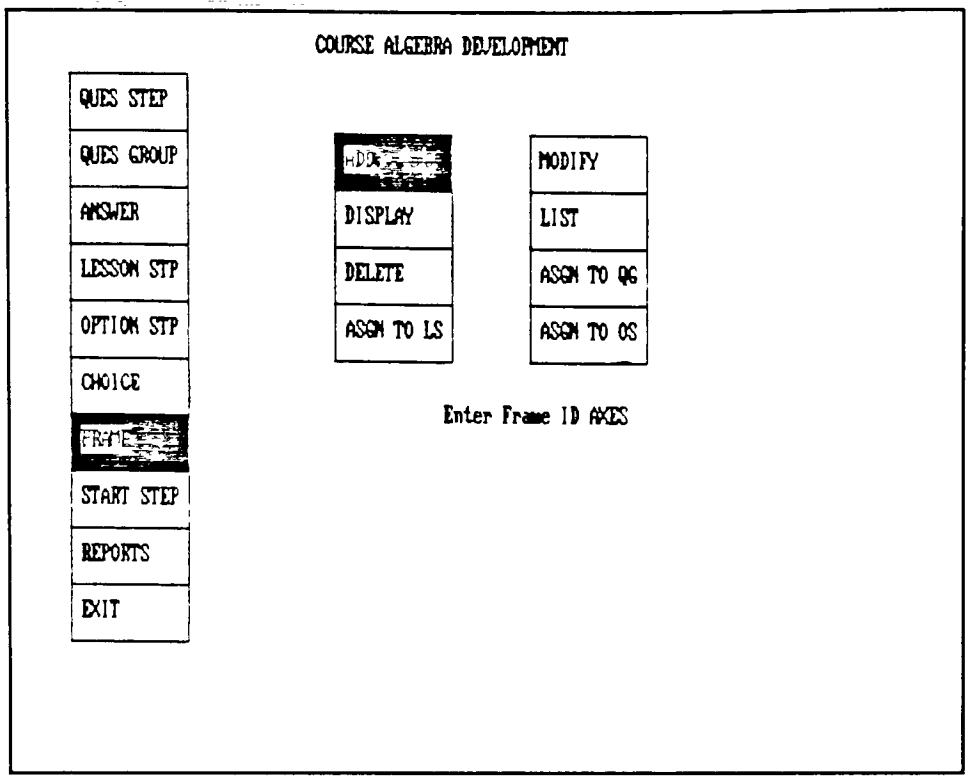
Authoring a course (AUTHOR.COM)

This section presents an example that builds a course named ALGEBRA and demonstrates a majority of the authoring capabilities. Appendix A contains the screens (including help screens) accessible to the author during an authoring session. It serves as a detailed description of the authoring capabilities.

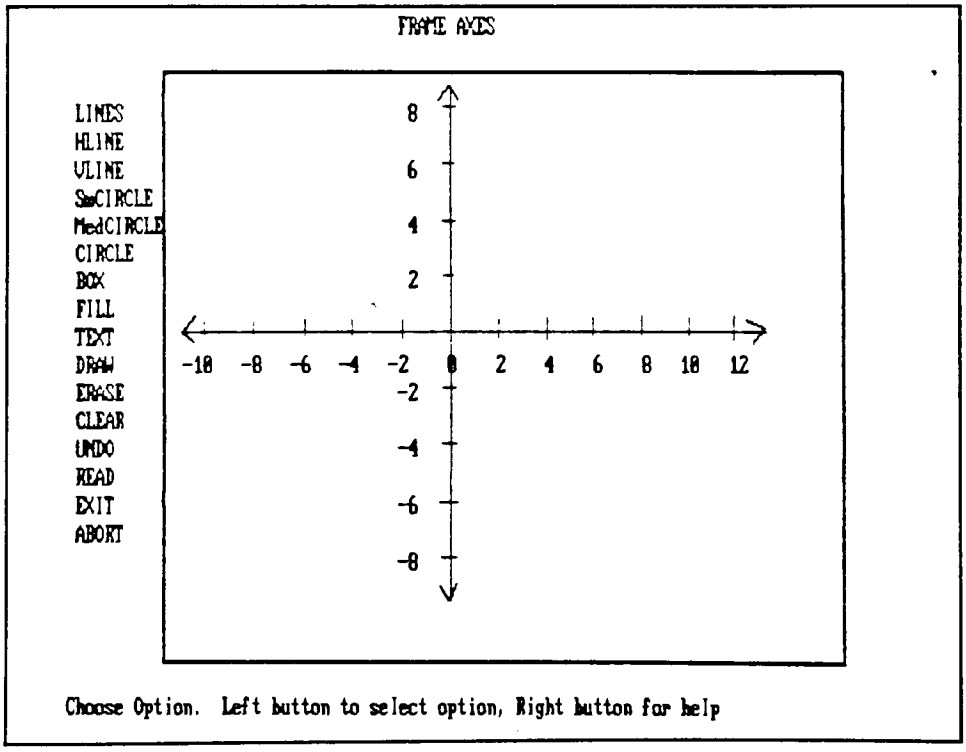


Type AUTHOR to initiate the running of the authoring system. Select the option ADD on the main menu with the mouse and respond to the Course Id prompt with ALGEBRA. This will initialize ALGEBRA as a new course and will proceed to the Course Development Menu.

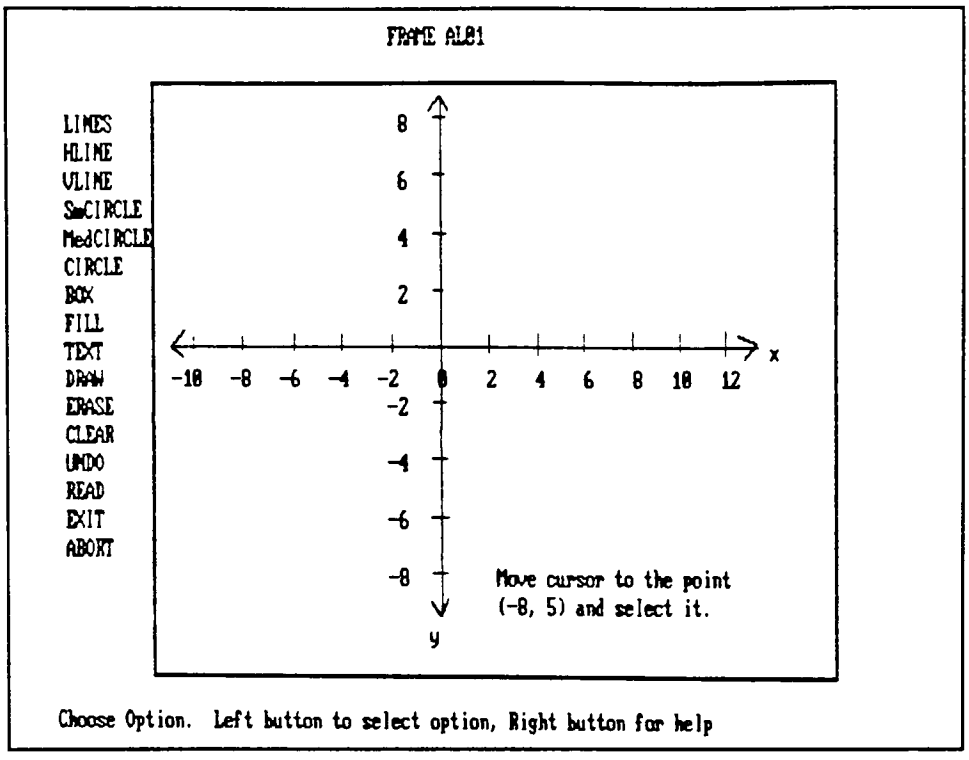
First, the question frames of the course will be developed. The author "draws" a frame by using the options on the left side of the frame development screen. Lesson and option frames, if they were implemented, would be created in the same way. The frames are the screen displays that will be seen by the student when taking the course.



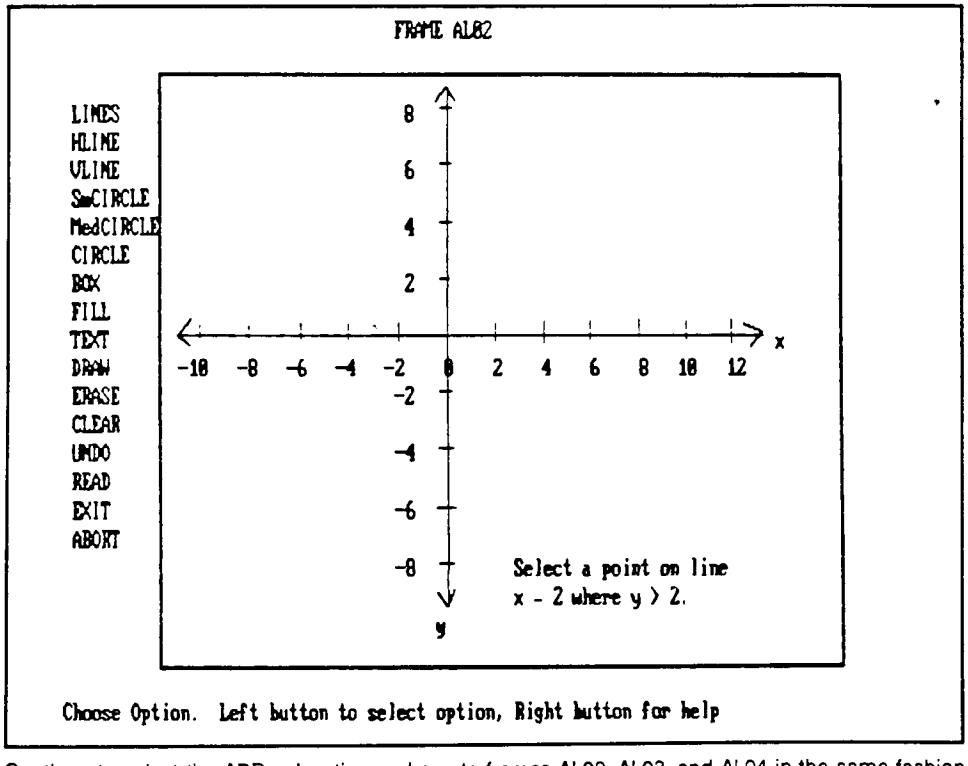
Select the FRAME option on the Course Development Menu. For each frame, select the ADD suboption and enter the Frame ID. The first frame to be added is AXES. This frame will be created to be used as a building block for creating some of the question frames of the course.



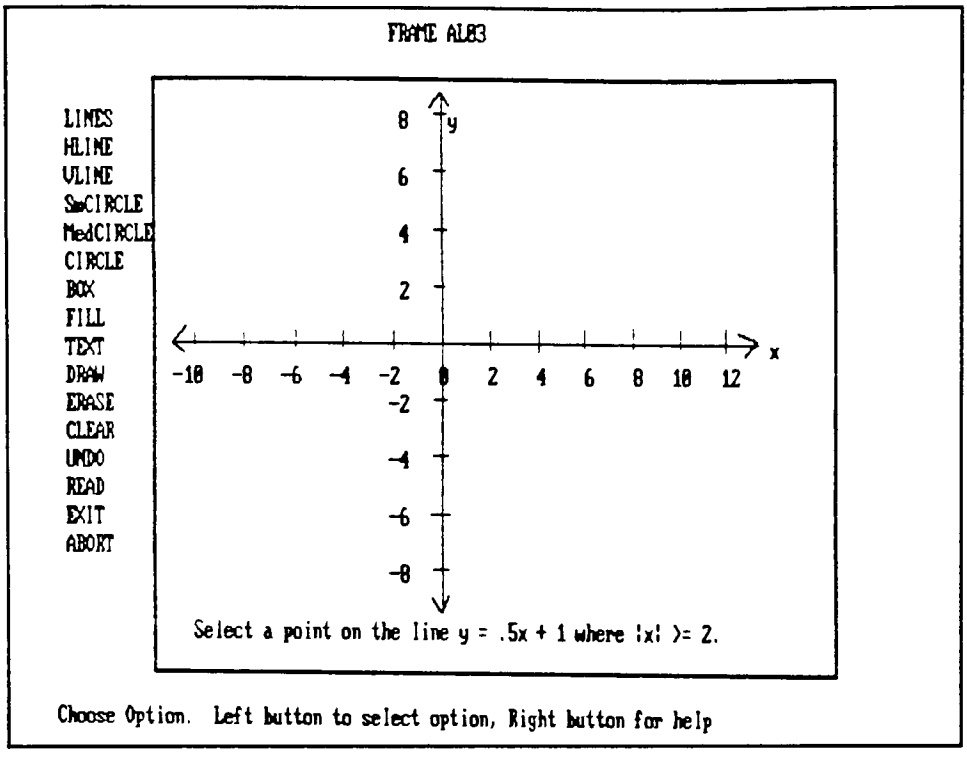
On the Frame Development screen, use the HLINE, VLINE, DRAW, and TEXT options to "draw" the frame. When completed, chose the EXIT option to save the frame and return to the previous menu.

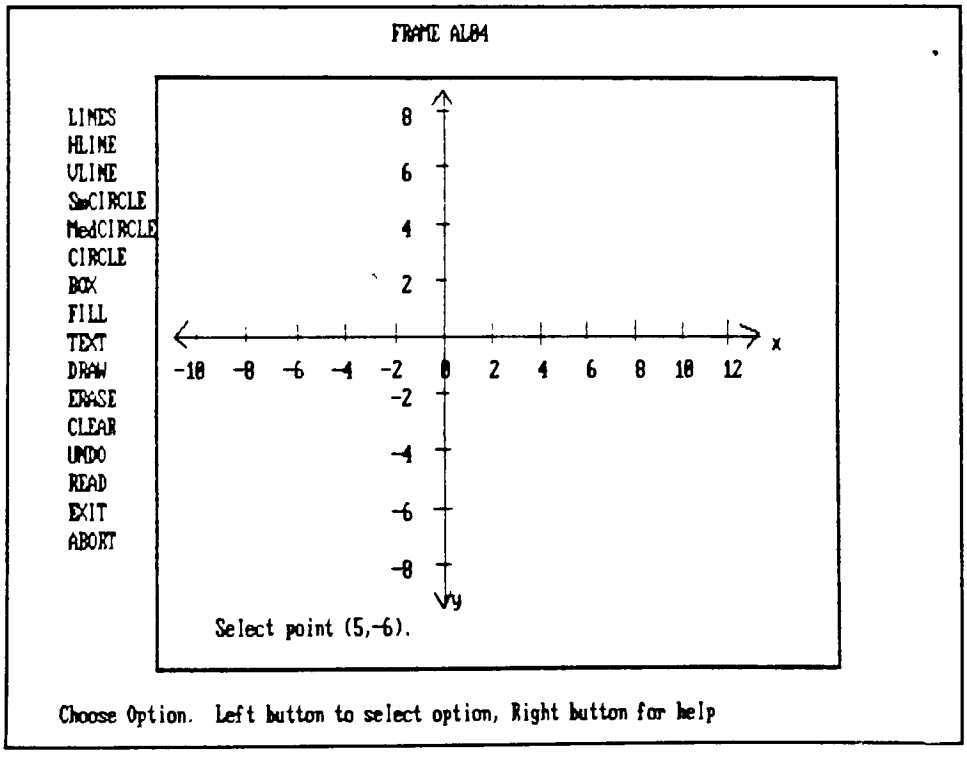


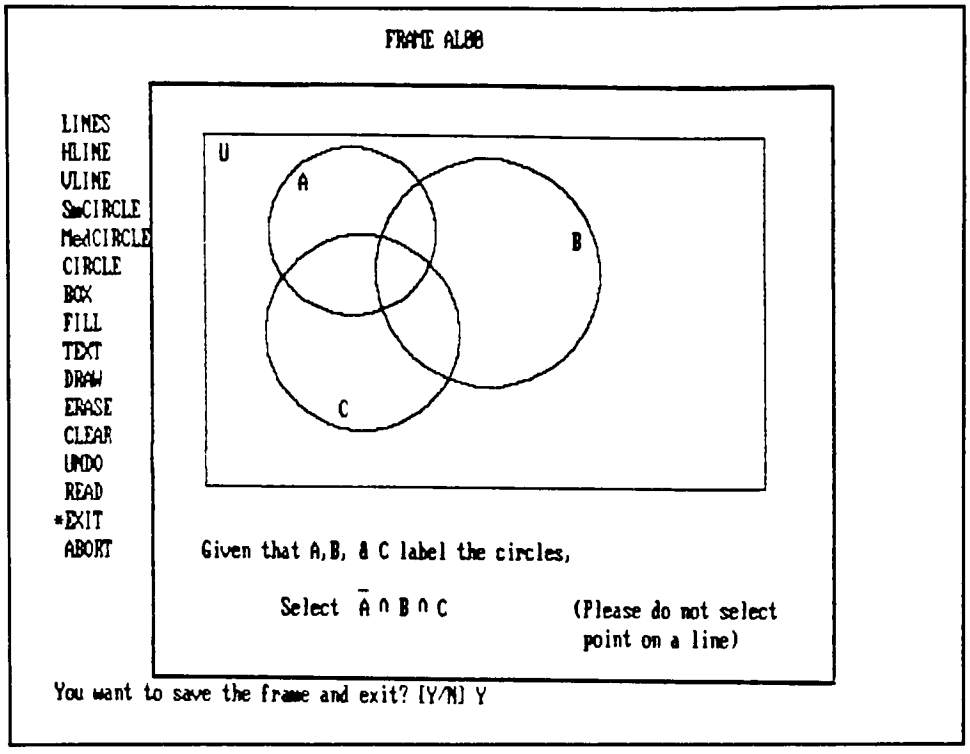
Select the ADD suboption and enter Frame ID of AL01. On the Frame Development screen, use the READ option to display the frame AXES that has already been drawn. Add the question text to the frame using the TEXT option. Select the EXIT option to save the frame and return to the previous menu.



Continue to select the ADD suboption and create frames AL02, AL03, and AL04 in the same fashion.


FRAME AL03
LINES
HLINE
VLINE
SmCIRCLE
MedCIRCLE
CIRCLE
BOX
FILL
TEXT
DRAW
ERASE
CLEAR
UNDO
READ
EXIT
ABORT
8
6
4
2
-2
-4
-6
-8
y
x
-10 -8 -6 -4 -2 0 2 4 6 8 10 12
Select a point on the line y = .5x + 1 where |x| >= 2.
Choose Option. Left button to select option, Right button for help


FRAME AL04
LINES
HLINE
VLINE
SmCIRCLE
MedCIRCLE
CIRCLE
BOX
FILL
TEXT
DRAW
ERASE
CLEAR
UNDO
READ
EXIT
ABORT
8
6
4
2
-2
-4
-6
-8
y
x
-10 -8 -6 -4 -2 0 2 4 6 8 10 12
Select point (5,-6).
Choose Option. Left button to select option, Right button for help

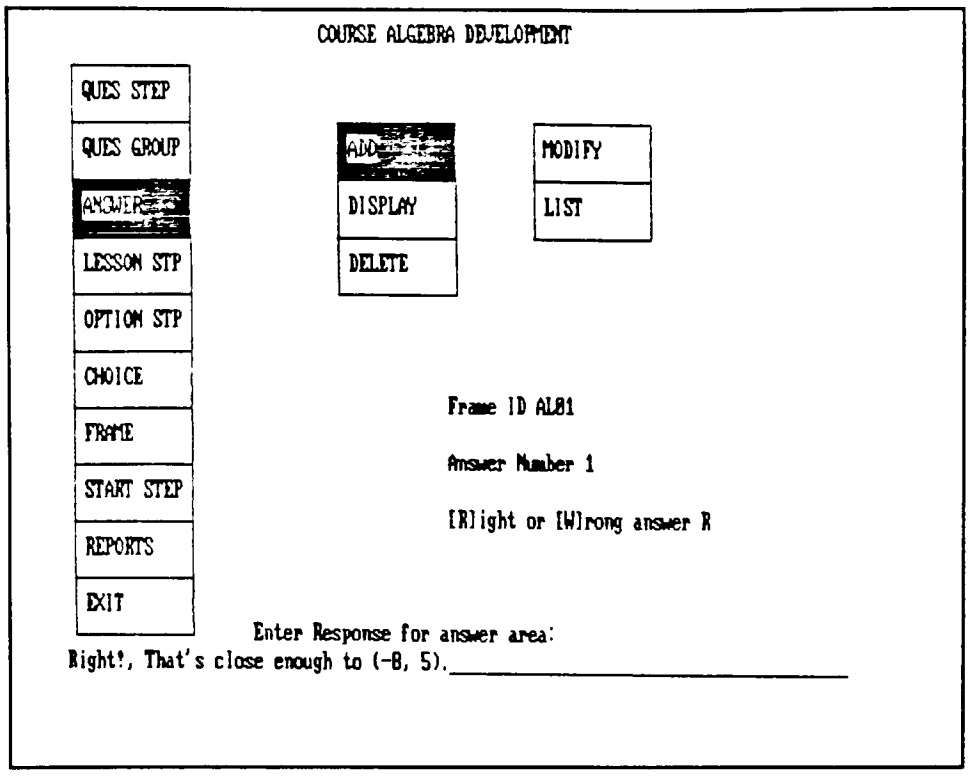


Select the ADD suboption to create frame AL00. On the Frame Development screen, use the CIRCLE, BOX, TEXT and ERASE options to draw the frame. (The intersection signs are the letter O with the bottom ERASEd.)

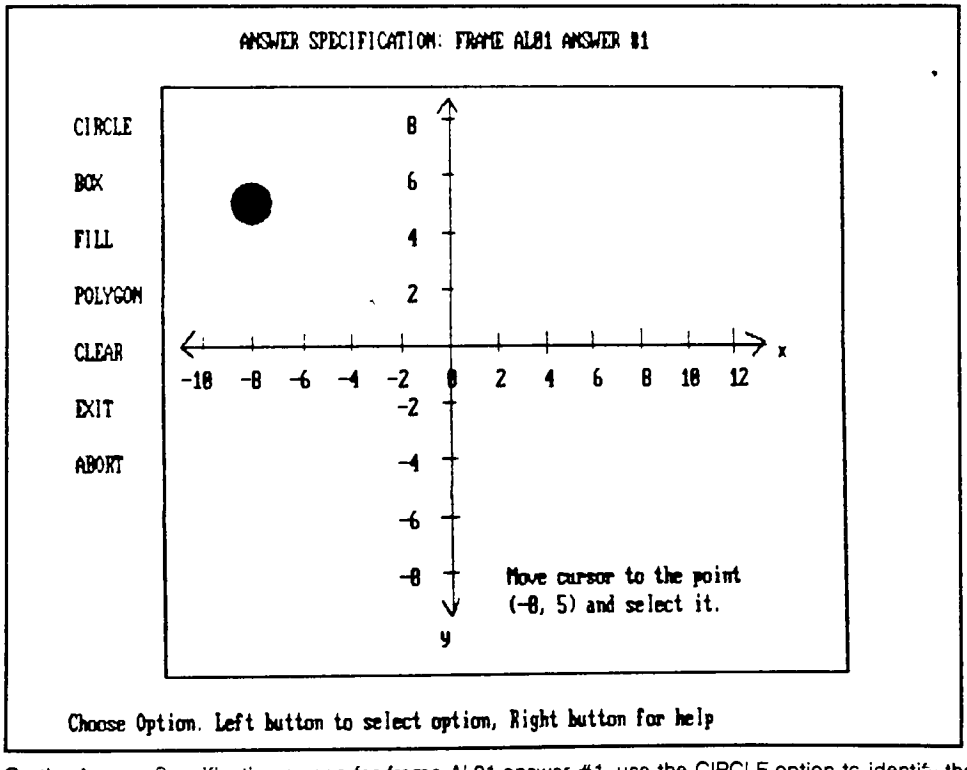
Now that all the question frames are created, answer specifications for them will be developed. An answer specification for a frame includes the answer number (0-5), whether the pixels within the answer area are right or wrong answer selections, the response to be displayed to the student if the student's answer selection is within the answer area, and the answer area itself (when the answer number is not 0).

Four options are available for identifying an answer area on the screen. The CIRCLE option allows for the specification of the area within a circle. The BOX option allows for the specification of a rectangular area. The FILL option allows the selection of a point within a fully bounded shape to identify the entire area within the bounded shape as the answer area. The author should be aware that text which is made up of lit pixels is considered to be a boundary and therefore will not be considered part of an answer area that is selected with the FILL option. The POLYGON option allows for the selection of the adjacent vertices of a polygon of up to ten sides. The area within in the polygon is the answer area.

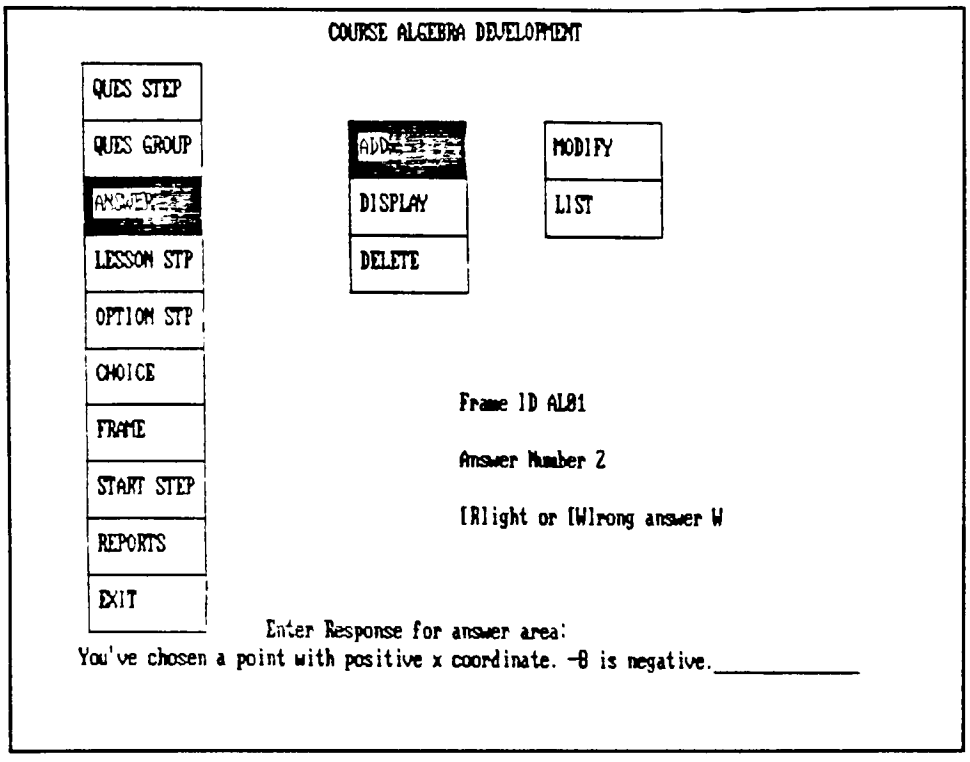
For one answer specification multiple areas can be selected and the union of the areas is the answer area. When answer number 0 is specified, no answer area is specified. The response that is specified in this case is the default response that is displayed to the student if the student's answer selection is not within any of the answer areas specified for the frame. If the author does not specify a default response, a system defined default response is displayed to the student.



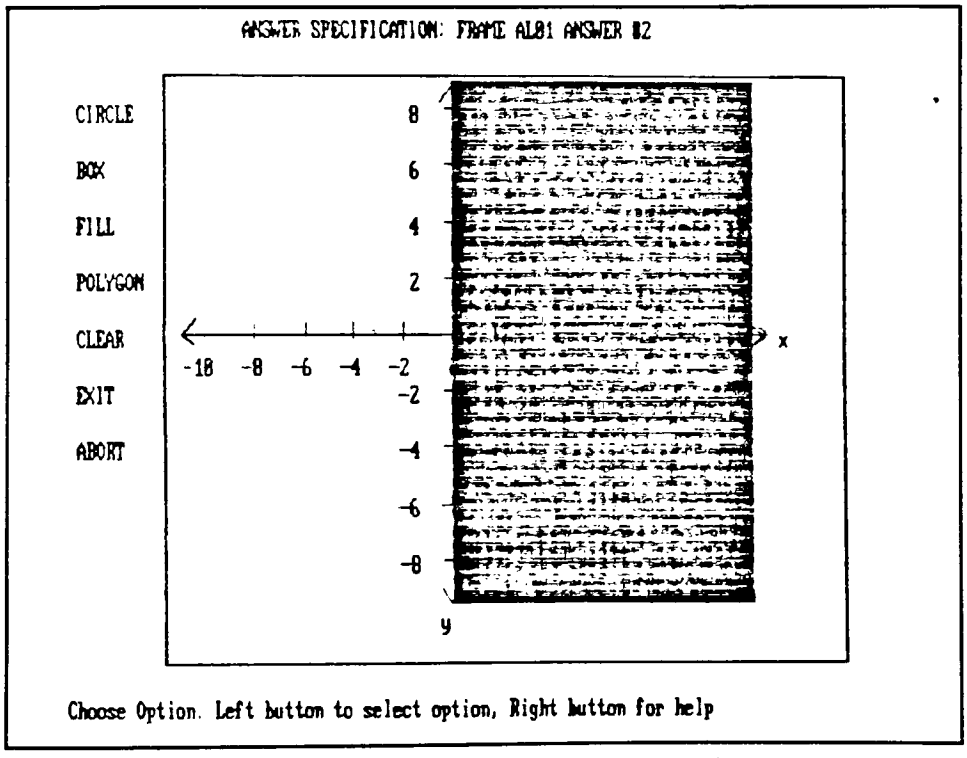
Select the ANSWER option on the Course Development Menu. Select the ADD suboption and enter Frame ID of AL01 and Answer Number of 1. Enter R to indicate that a right answer is going to specified and enter "Right!, That's close enough to (-8, 5)." as the response to the student for this answer selection.



On the Answer Specification screen for frame AL01 answer #1, use the CIRCLE option tc identify the area around the point as acceptable answer #1 selections. The selected circle is filled in. Select the EXIT option to save the answer specification and return to the previous menu.

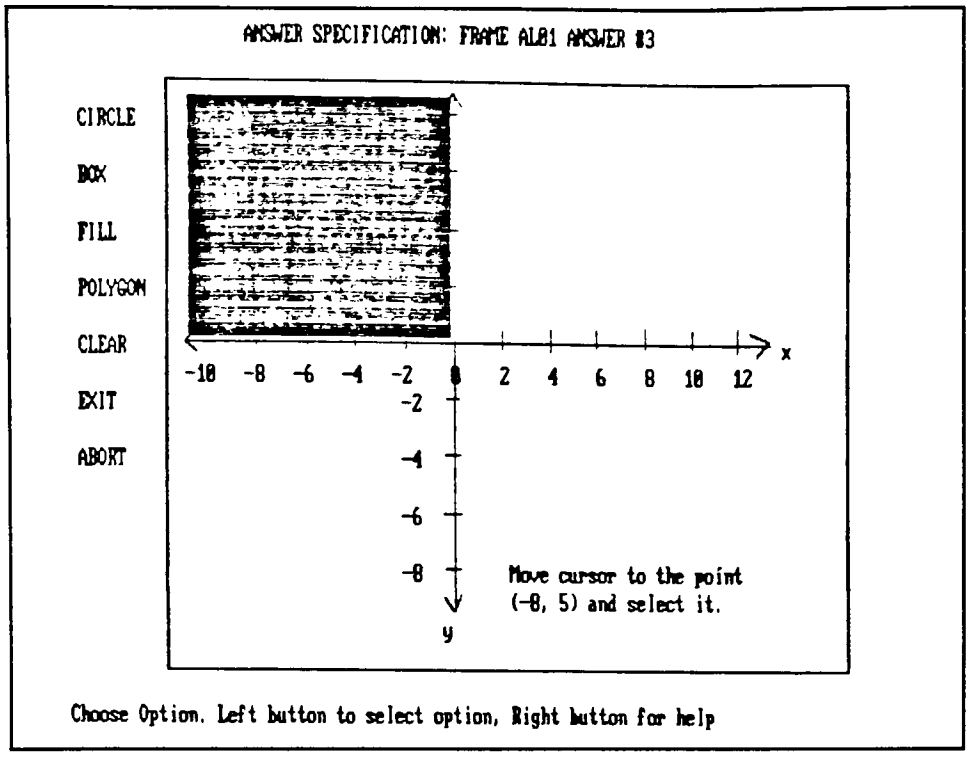


Select the ADD suboption again. Enter Frame ID of AL01 and Answer Number of 2. Then enter W to indicate a wrong answer area is to be specified and enter "You've chosen a point with a positive x coordinate. -8 is negative." as the response to the student when the answer area is selected.

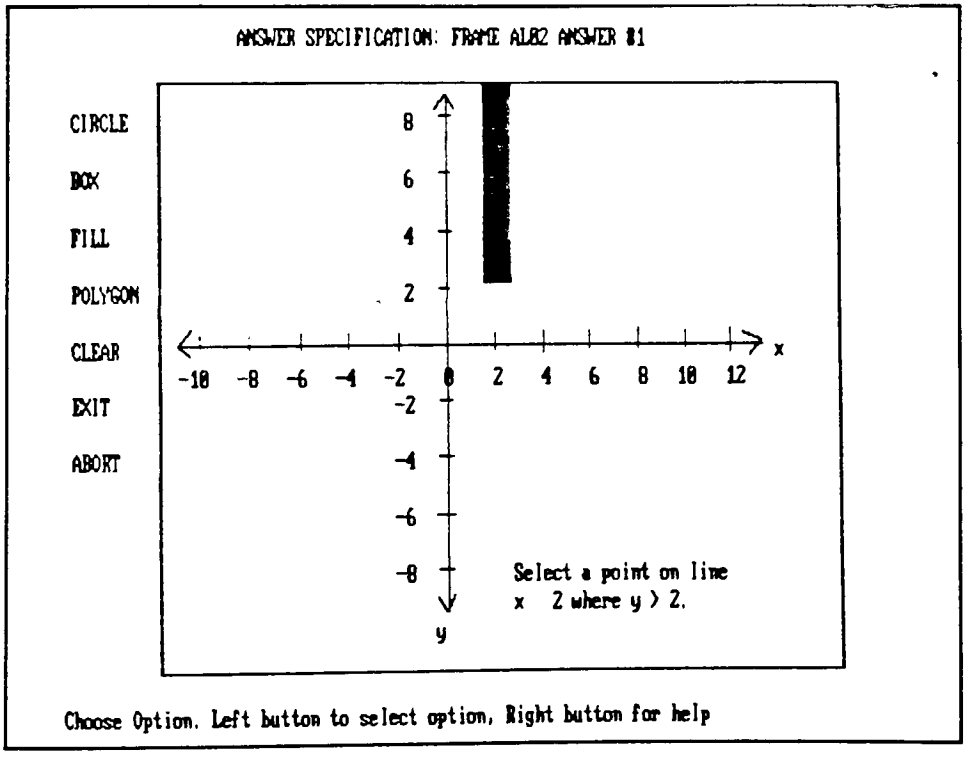


On the Answer Specification screen for frame AL01 answer #2, use the BOX option to identify a rectangular area as the answer area. The selected rectangle is filled in. Select the EXIT option to save the answer specification and return to the previous menu.

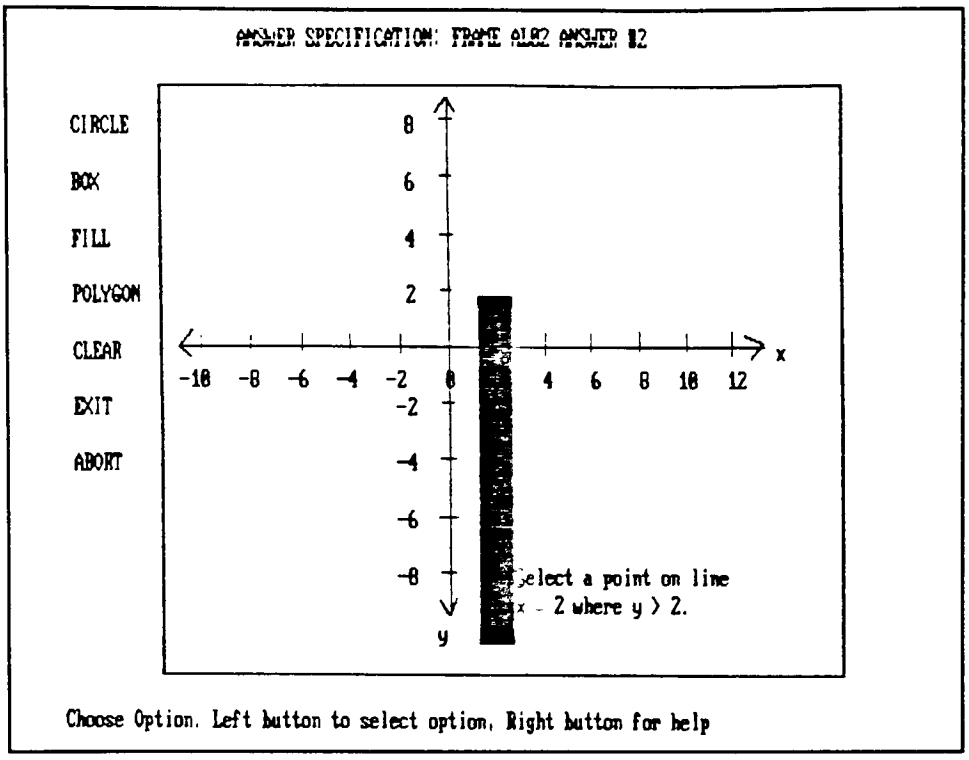
Continue to select the ADD suboption to make the following answer area specifications:



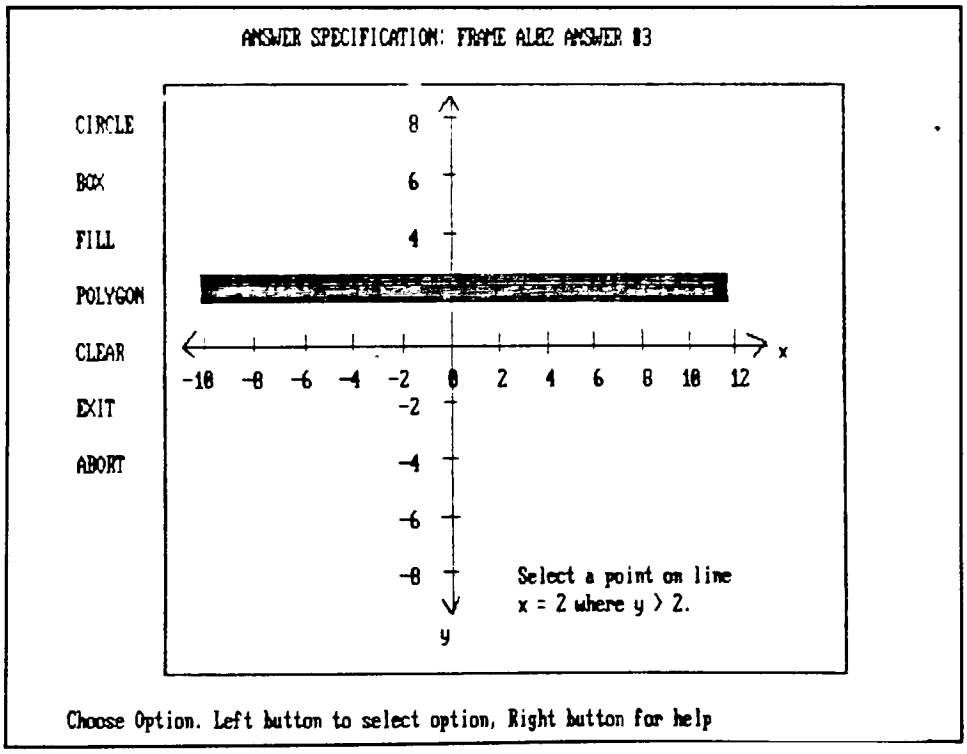
Specify Answer Number 3 for frame AL01 as a wrong answer with the response of "You're not close enough to the point, but you're in the correct quadrant". On the Answer Specification screen use the BOX option to specify the area.



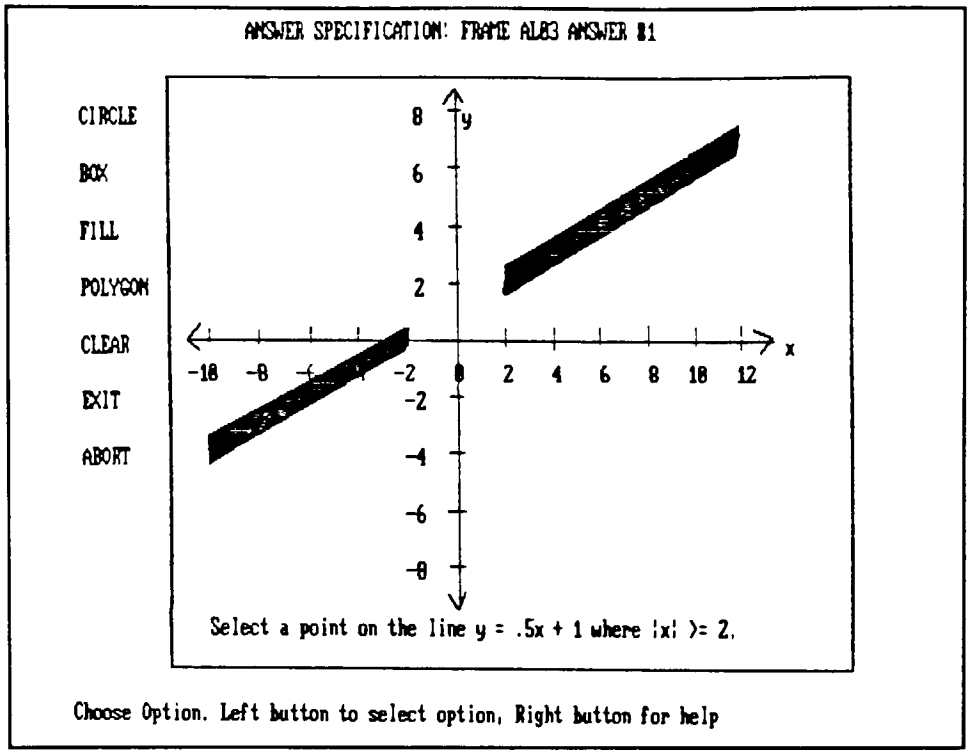
Specify Answer Number 1 for frame AL02 as a right answer with the response of "Correct! You're on the line.". On the Answer Specification screen, use the BOX option to specify the answer area.



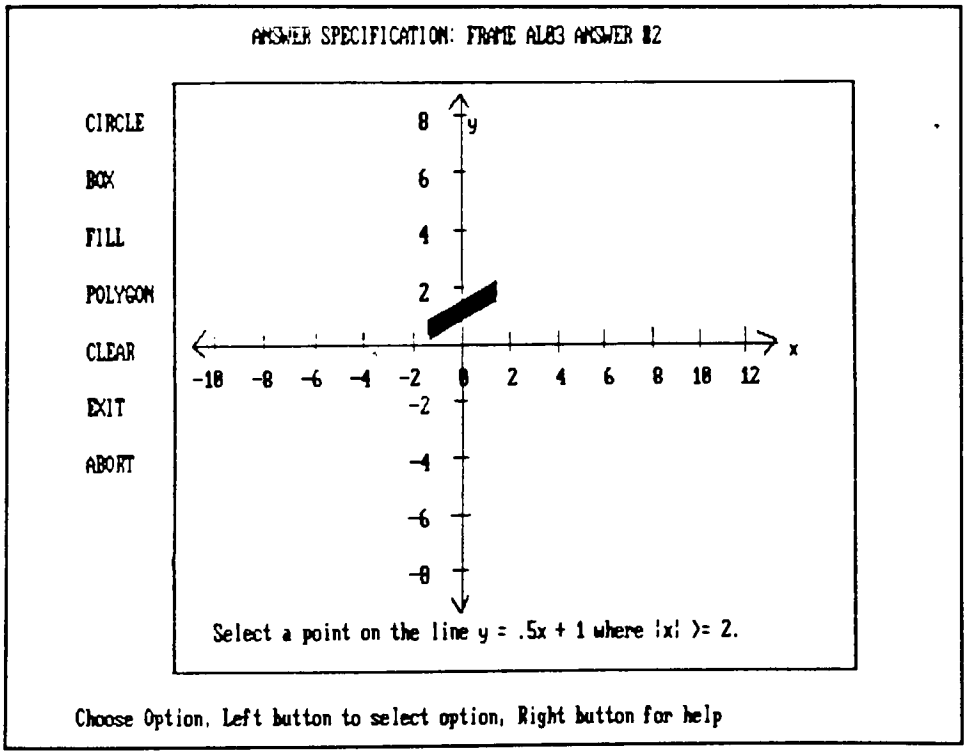
Specify Answer Number 2 for frame AL02 as a wrong answer with the response of "You're on line x = 2 but y coordinate is not greater than 2.". On the Answer Specification screen, use the BOX option to specify the answer area.



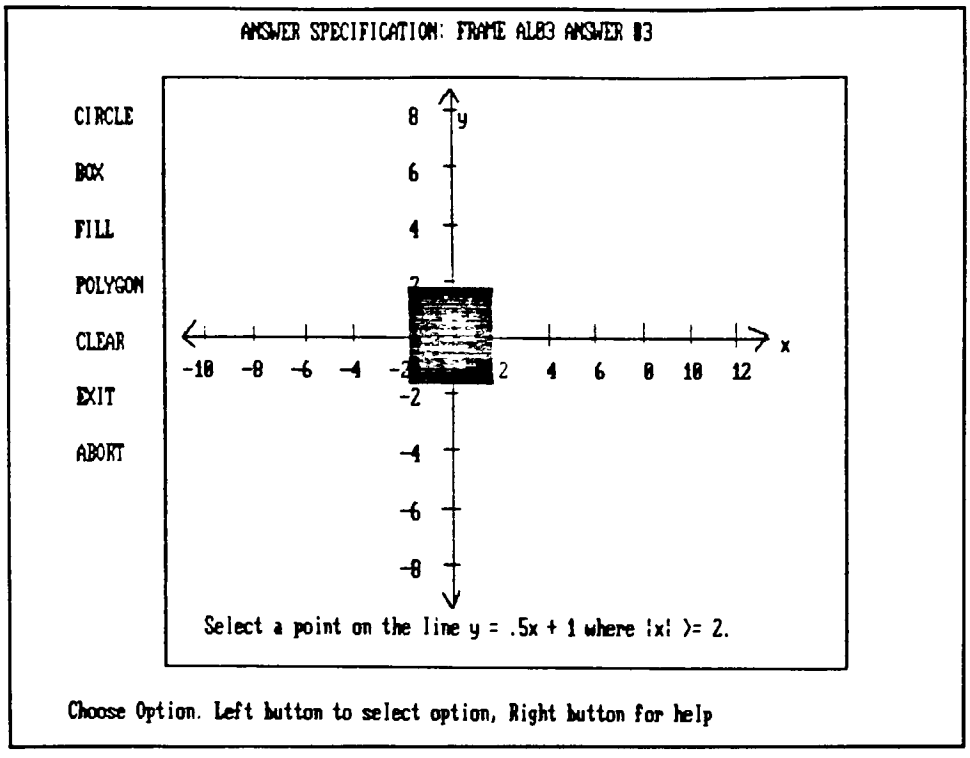
Specify Answer Number 3 for frame AL02 as a wrong answer with the response of "You're on the line y = 2. You want to be on line x = 2.". On the Answer Specification screen, use the BOX option to specify the answer area.



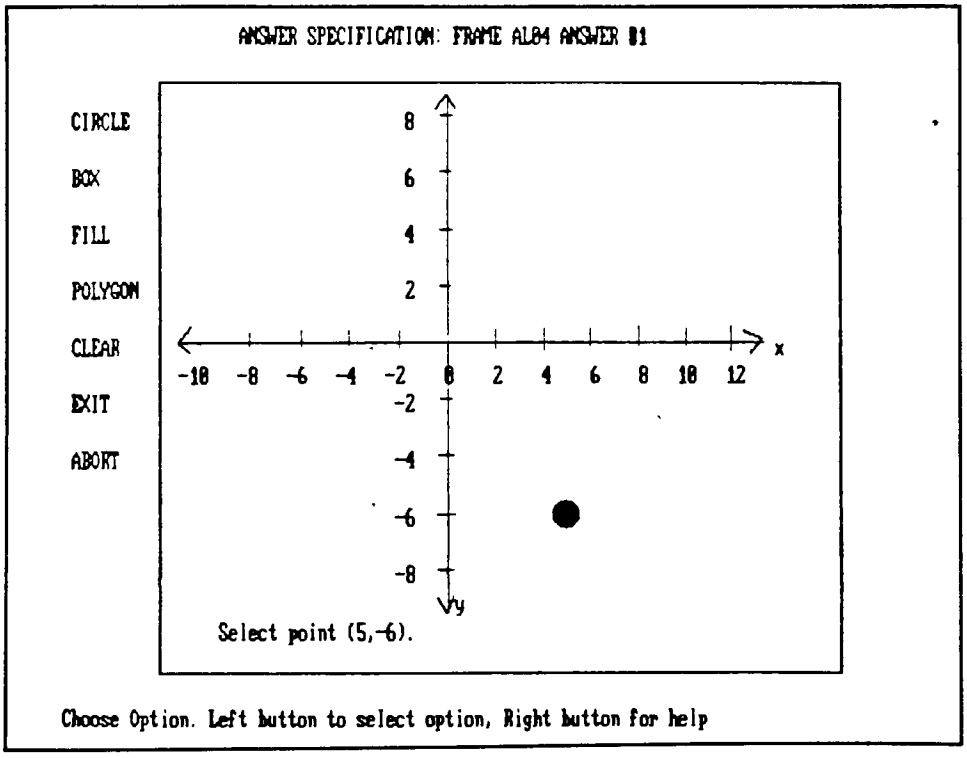
Specify Answer Number 1 for frame AL03 as a right answer with the response of "Good! You've selected a correct point.". On the Answer Specification screen, use the POLYGON option multiple times to specify the answer areas.



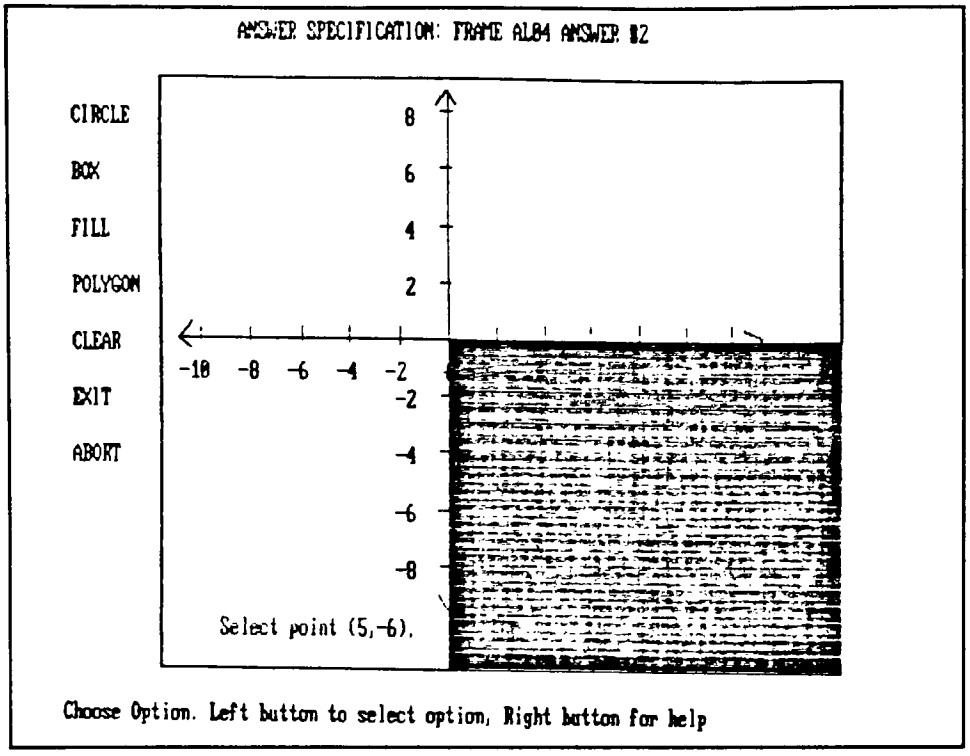
Specify Answer Number 2 for frame AL03 as a wrong answer with the response of "You're on y = .5x + 1, but the selected x coordinate is less than 2.". On the Answer Specification screen, use the POLYGON option to specify the answer area.



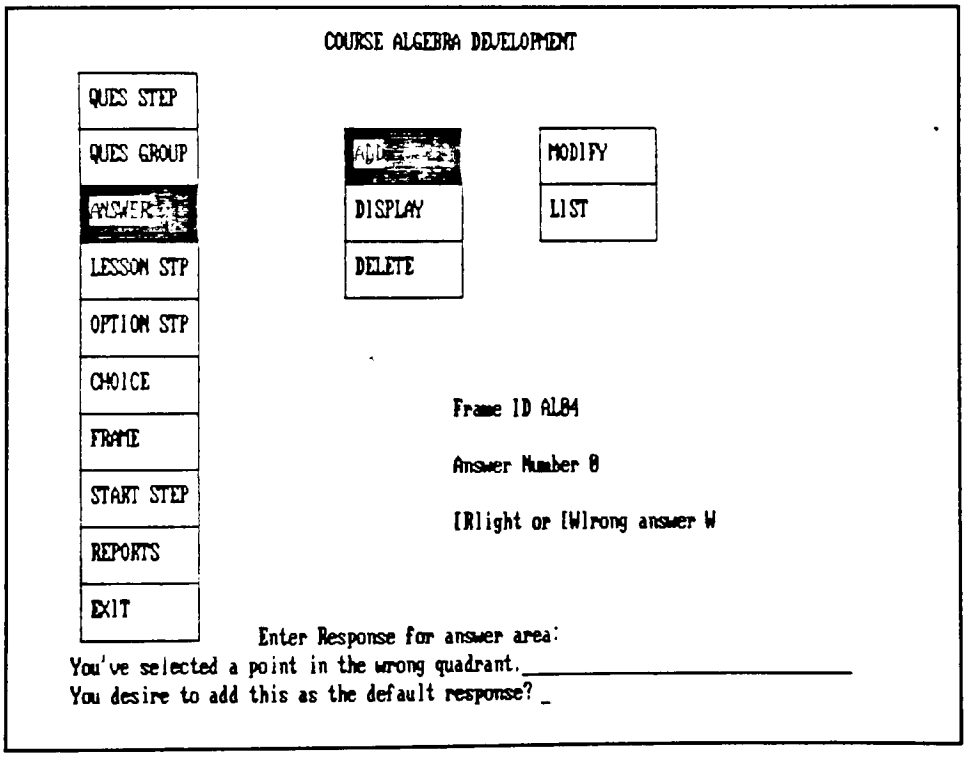
Specify Answer Number 3 for frame AL03 as a wrong answer with the response of "The absolute value of the x coordinate of the selected point is less than 2." On the Answer Specification screen, use the BOX option to specify the answer area.



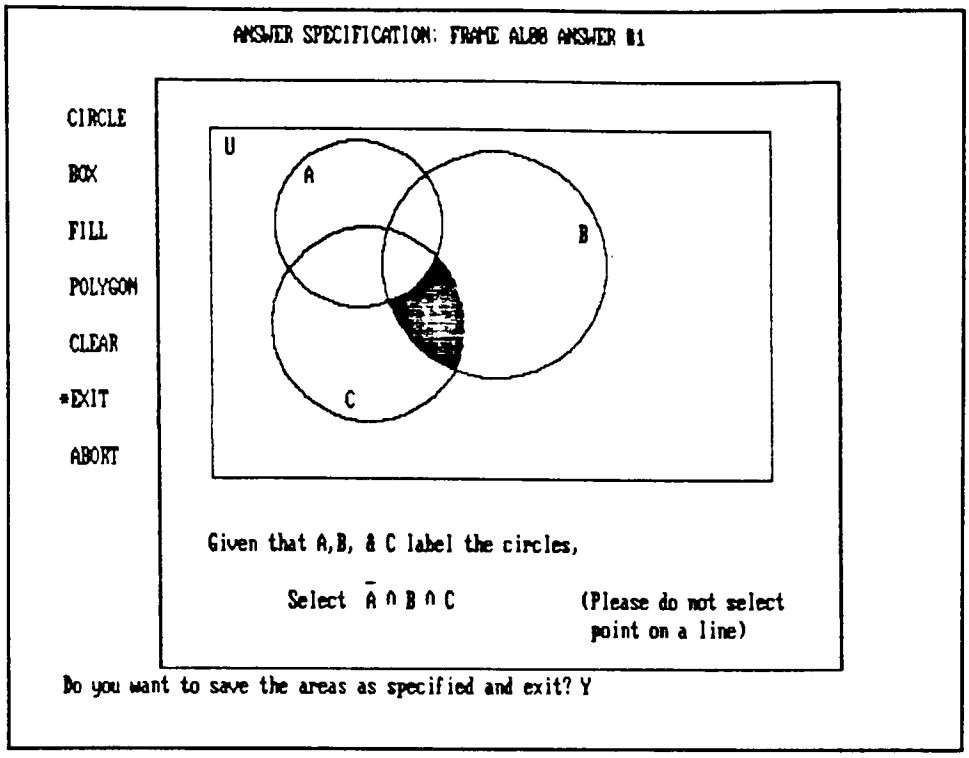
Specify Answer Number 1 for frame AL04 as a right answer with the response of "Correct! That's the point.". On the Answer Specification screen, use the CIRCLE option to specify the answer area.



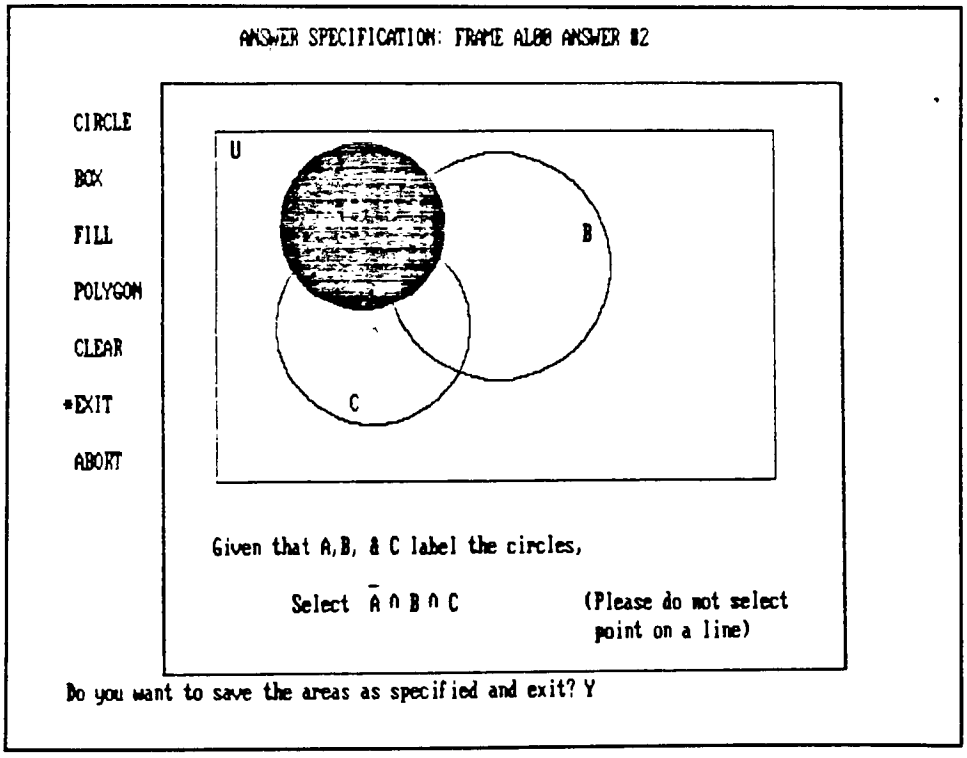
Specify Answer Number 2 for frame AL04 as a wrong answer with the response of "You're in the correct quadrant but not close enough to the point.". On the Answer Specification screen, use the BOX option to specify the answer area.



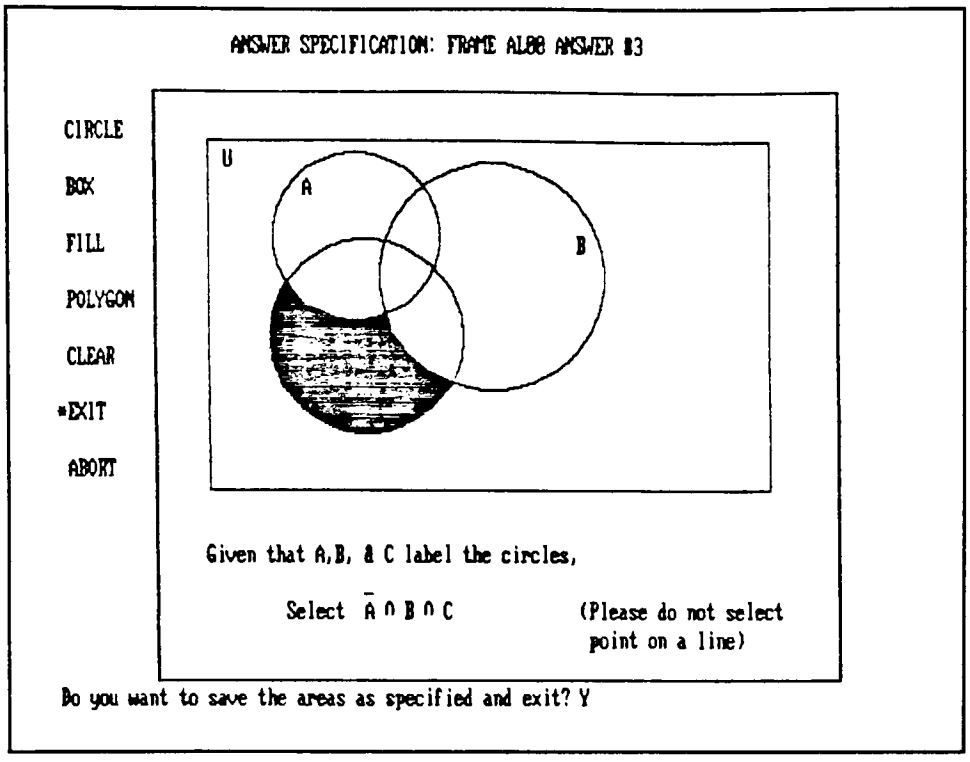
To specify a default answer for frame AL04, specify Answer Number 0 and the response. This is the response that is given if a point that is not within any of the specified answer areas is chosen by the student. When the author does not specify a default response, the system default response is given to the student.



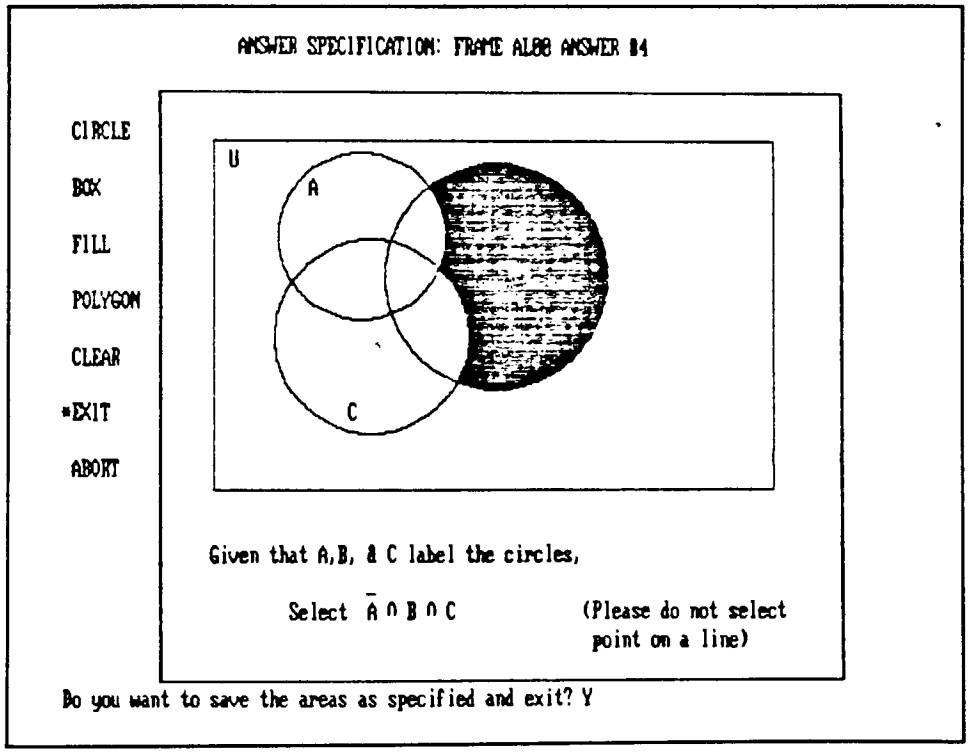
Specify Answer Number 1 for frame AL00 as a right answer with the response of "Right! That point is not in A and is in both B and C.". On the Answer Specification screen, use the FILL option to specify the answer area. The FILL option allows you to choose a bounded shape as the area by pointing to a point within the shape.



Specify Answer Number 2 for frame AL00 as a wrong answer with the response of "The selected point is within A, The desired area does not include A.". On the Answer Specification screen, use the BOX and FILL options to specify the answer area. Put a BOX around the letter A first so that if the student points to the lit pixels of letter A it will be considered within the area. Then select the rest of circle A with the FILL option.

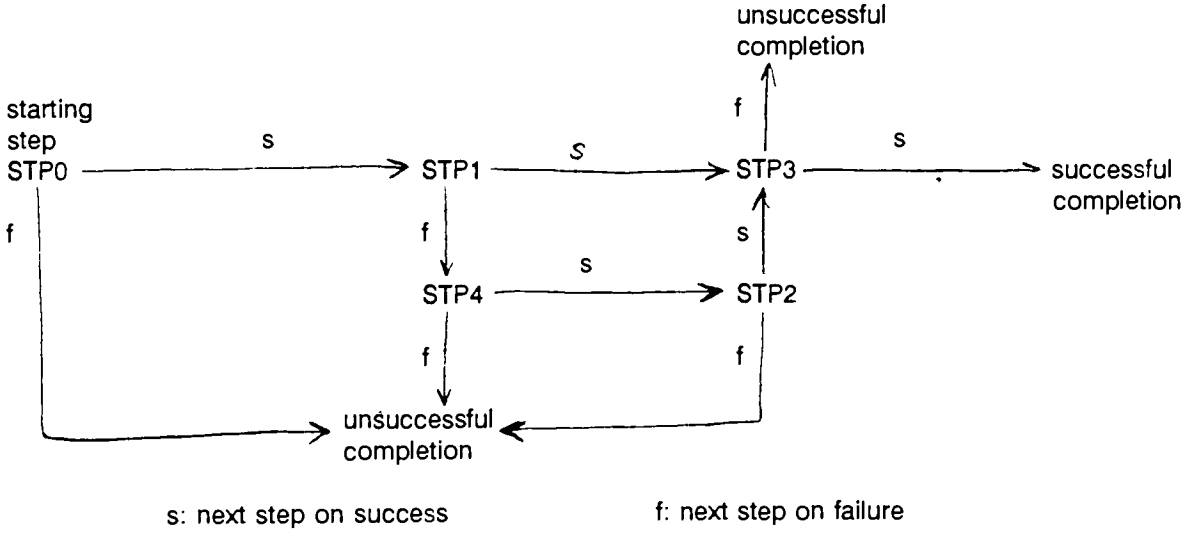


Specify Answer Number 3 for frame AL00 as a wrong answer with the response of "The point selected is not within B." On the Answer Specification screen, use the BOX and FILL options as in the the above specification.



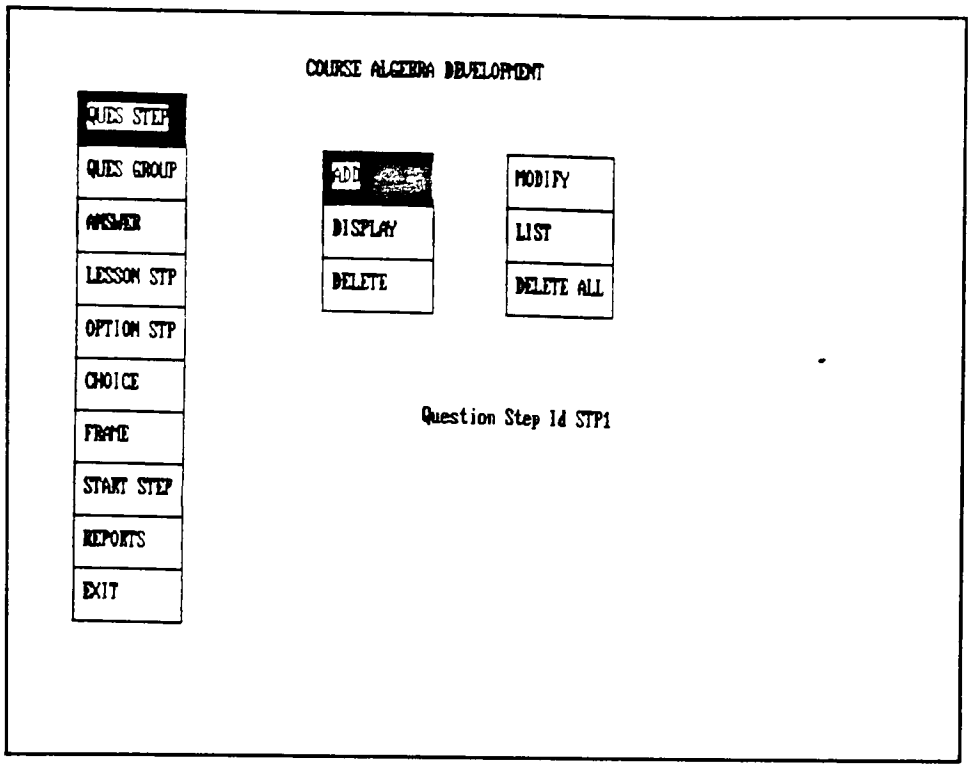
Specify Answer Number 4 for frame AL00 as a wrong answer with the response of "That's not a point in the desired area since it is not within C.". On the Answer Specification screen, use the BOX and FILL options as in the above specification.

The answer specifications are now complete. The sequencing of the steps of the course will be identified by adding the question step descriptions. This activity can be done before frame and answer area specification if desired. The question step description includes the identification of the frame that is displayed when the step is reached during the presentation of the course to the student. The performance criteria values of correct answers required, minimum questions asked, maximum questions asked, and the threshold number are all set to 1 by the system since question groups are not included in this implementation. These values identify how well the student must perform on the group of questions in order to successfully complete the step. The maximum tries entered by the author specifies the number of tries the student is allowed to correctly answer the question. The success step and failure step specify the next step to be presented to the student dependent on whether the step was successfully completed. (That is, whether the question was answered correctly within the maximum number of tries.) The steps for this course are sequenced as shown in the following diagram. All steps are question steps.

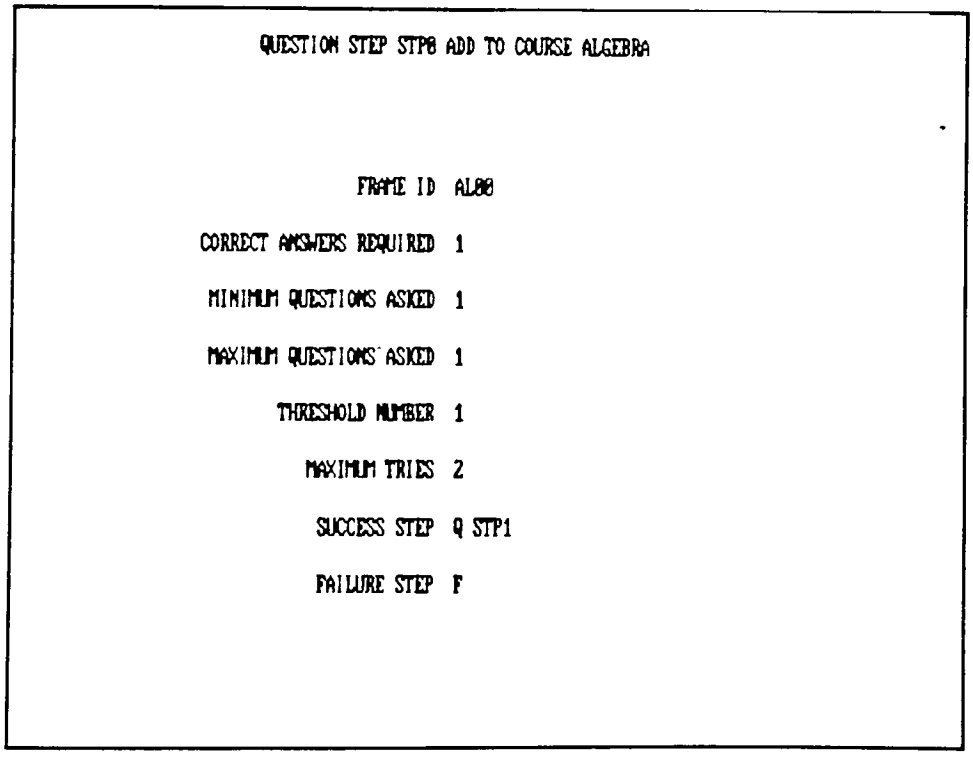


s: next step on success

f: next step on failure



Select the QUES STEP option on the Course Development Menu and choose the ADD suboption. Respond to Step ID with STP1 and you will proceed to the Question Step Add screen.



Enter the FRAME ID, MAXIMUM TRIES, SUCCESS STEP and FAILURE STEP as shown on the screen. AL00 is the frame that is displayed when this step of the course is encountered. The student gets 2 chances to answer the questions correctly. If the question is answered correctly, the student proceeds to question (Q) step STP1 of the course. If the student does not answer correctly the course is concluded with an unsuccessful status. (When S is specified, the course is concluded with a successful status.)

```
QUESTION STEP STP1 ADD TO COURSE ALGEBRA

                FRAME ID  AL81
CORRECT ANSWERS REQUIRED  1
 MINIMUM QUESTIONS ASKED  1
 MAXIMUM QUESTIONS ASKED  1
       THRESHOLD NUMBER   1
          MAXIMUM TRIES   2
           SUCCESS STEP   Q STP3
           FAILURE STEP   Q STP4
```

The remainder of the steps of the course are specified as shown in this and following screens.

```
QUESTION STEP STP2 ADD TO COURSE ALGEBRA

                FRAME ID  AL82
CORRECT ANSWERS REQUIRED  1
 MINIMUM QUESTIONS ASKED  1
 MAXIMUM QUESTIONS ASKED  1
       THRESHOLD NUMBER   1
          MAXIMUM TRIES   2
           SUCCESS STEP   Q STP3
           FAILURE STEP   F
```

QUESTION STEP STP3 ADD TO COURSE ALGEBRA

FRAME ID AL83

CORRECT ANSWERS REQUIRED 1

MINIMUM QUESTIONS ASKED 1

MAXIMUM QUESTIONS ASKED 1

THRESHOLD NUMBER 1

MAXIMUM TRIES 3

SUCCESS STEP S

FAILURE STEP F

QUESTION STEP STP4 ADD TO COURSE ALGEBRA

FRAME ID AL84

CORRECT ANSWERS REQUIRED 1

MINIMUM QUESTIONS ASKED 1

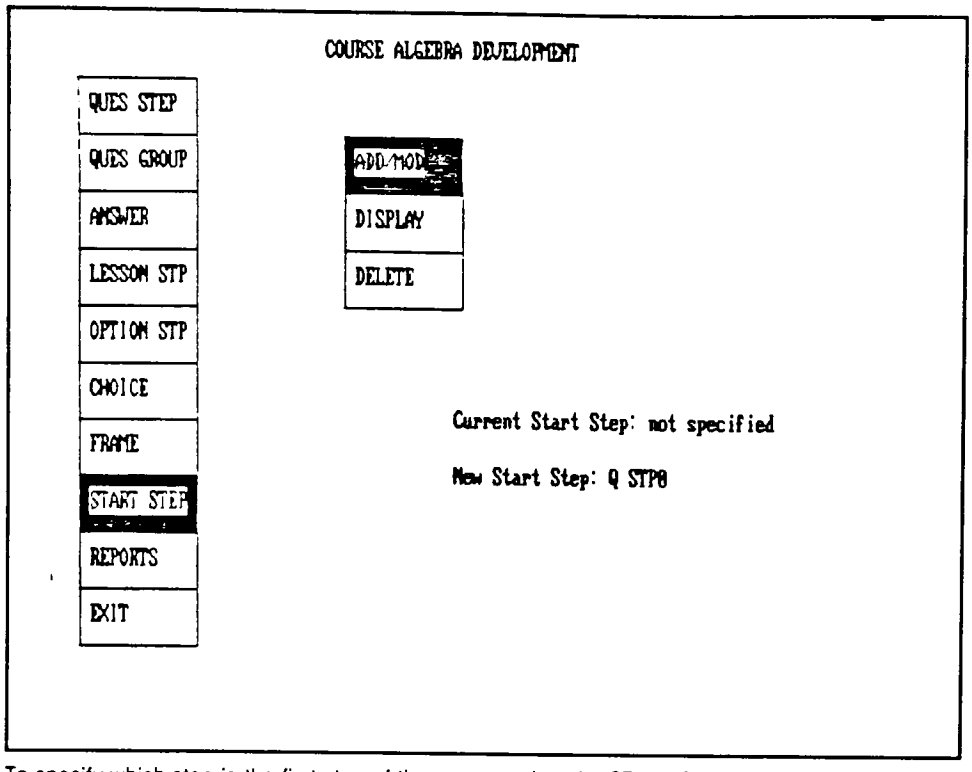
MAXIMUM QUESTIONS ASKED 1

THRESHOLD NUMBER 1

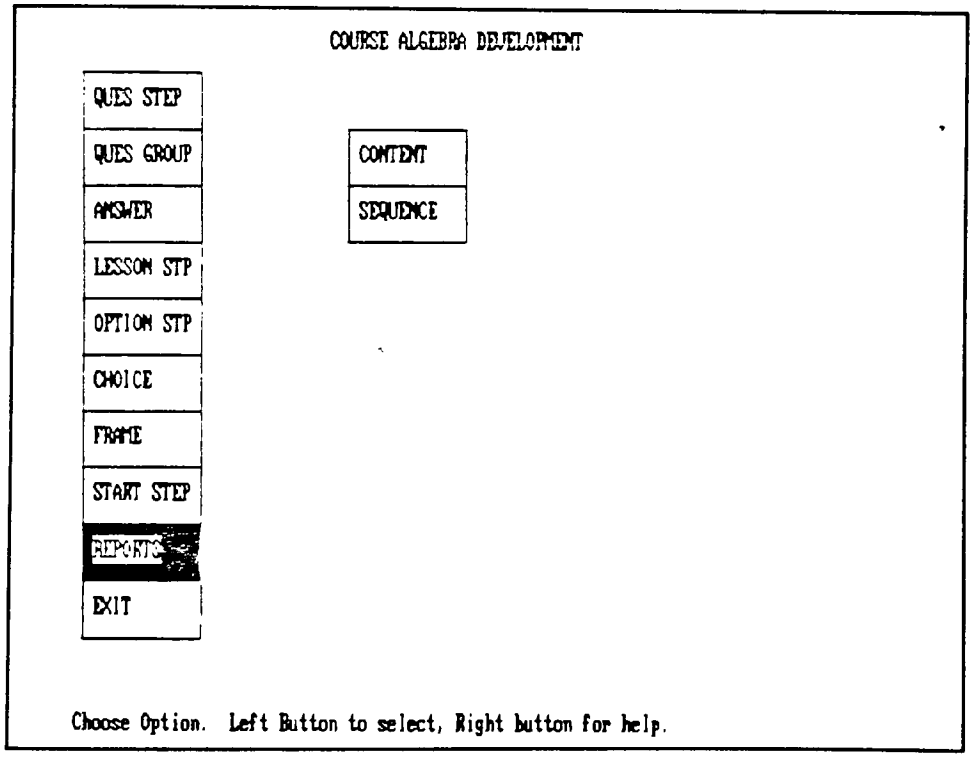
MAXIMUM TRIES 2

SUCCESS STEP Q STP2

FAILURE STEP F



To specify which step is the first step of the course, select the START STEP option on the Course Development Menu. Select the ADD/MOD suboption and enter the Step Type of Q and the Step ID of STP0.



To validate the completeness of the course. Select the REPORTS option on the Course Development Menu.

```
                    *** CONTENT REPORT ***

                      Course ID: ALGEBRA


*** QUESTION STEPS ***

     Question                          Max#      Min#      Max#    Required      Last
     Step ID        Frame ID         Tries     Asked     Asked    Correct     Asked

       STP0           AL00             2         1         1         1         1
       STP1           AL01             2         1         1         1         1
       STP2           AL02             2         1         1         1         1
       STP3           AL03             3         1         1         1         1
       STP4           AL04             2         1         1         1         1


** FRAMES ***

     FRAME       Undef      USED BY STEP:

       AL00                 Question Step STP0
       AL01                 Question Step STP1
       AL02                 Question Step STP2
       AL03                 Question Step STP3
       AL04                 Question Step STP4


END OF REPORT
```

Select the CONTENT suboption to produce this Content Report. The report does not indicate any Incompleteness since all the question step information is specified and none of the frames have an undefined flag in the 'Undef' column.

```
                    *** SEQUENCE REPORT ***
                      Course ID: ALGEBRA

                 Start Step: Question Step STP0

***Undefined Steps***


***QUESTION STEPS***

  Step
  Number       Next On Success:       Next On Failure:      Unreachable

    STP0       Question Step STP1       Failure End
    STP1       Question Step STP3     Question Step STP4
    STP2       Question Step STP3       Failure End
    STP3        Success End             Failure End
    STP4       Question Step STP2       Failure End


END OF REPORT
```

Select the SEQUENCE suboption to produces this Sequence Report. The report does not indicate any incompleteness since no steps are listed under the "Undefined Steps" and none of the steps are flagged in the "Unreachable column". If a step is flagged with a asterisk in the "Unreachable" column it indicates that the step could not be reached as the step sequencing is currently specified.

Select the EXIT option on the Course Development Menu and the LEAVE option on the MAIN MENU to complete the authoring session.

Taking a course (STUDENT.COM)

The following sequence of screens is an example of taking the course ALGEBRA that was developed in the previous description. The small rectangle displayed on the screen marks the point selected by the student.

After the student has identified the course to be taken, the frame assigned to the first step of the course is displayed. A directive message always appears on the last line of the screen to indicate to the student what action the system is waiting for. Each time the student selects an answer it is evaluated to identify if it's within one of the areas specified by the author. If it's inside one of the specified areas, the response that the author identified for the area is displayed on the second last line of the screen. The default response is displayed if the student's selected answer is not within any of the answer areas.

When the question is answered correctly, the system continues to the step of the course that the author specified as the next step on success. If the question is answered incorrectly and the student has not tried the maximum number of tries specified for the step, the system asks the student to try again. When the student has failed to answer the question correctly within the maximum number of tries, the system will highlight all the areas that the author specified as right answer areas and then continue to the step that was specified as the next step on failure. The presentation of the course will continue in this fashion until the next step specified is either the success or failure end code.

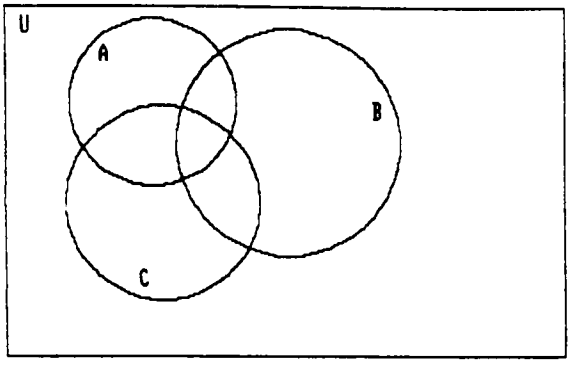
WELCOME to GAS Course Delivery

Enter Student ID: STUDNT2

Press F10 to exit

Enter Course ID ALGEBRA

AL88

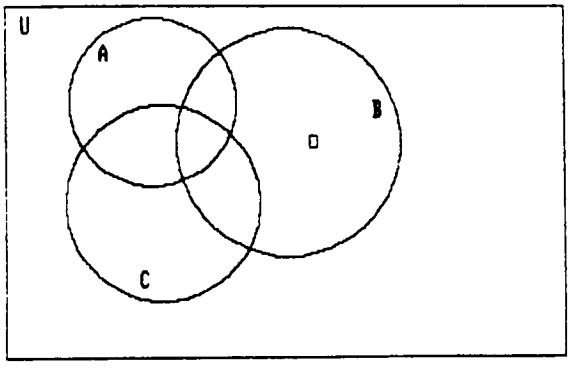


Given that A,B, & C label the circles,

Select $\bar{A} \cap B \cap C$ (Please do not select point on a line)

Waiting for answer selection

AL88

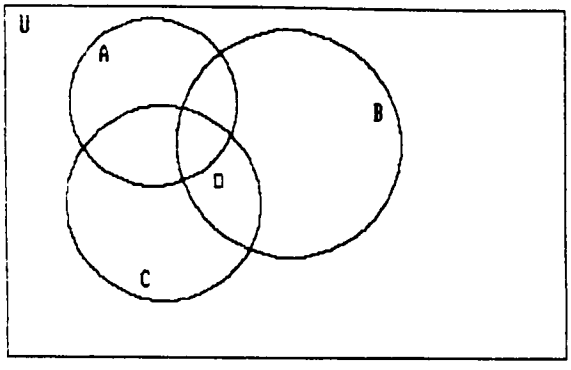


Given that A,B, & C label the circles,

Select $\bar{A} \cap B \cap C$ (Please do not select point on a line)

That's not a point in the desired area since it is not within C.
Try again. Select answer

AL00

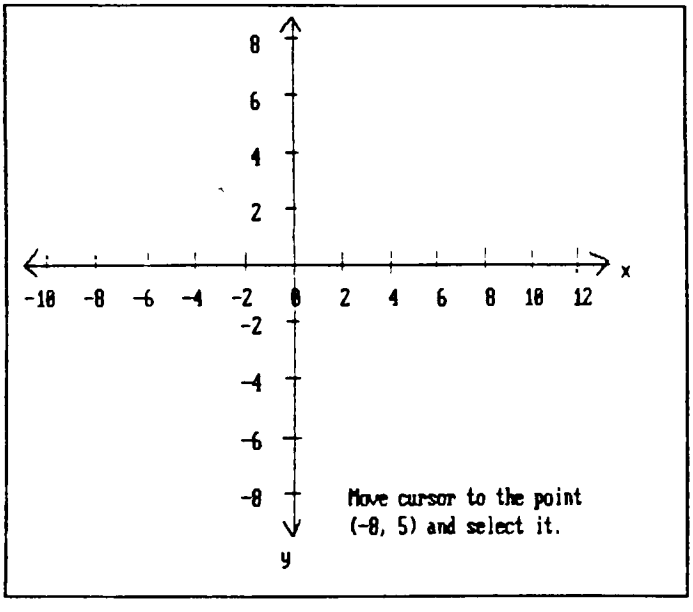


Given that A,B, & C label the circles,

Select $\bar{A} \cap B \cap C$ (Please do not select point on a line)

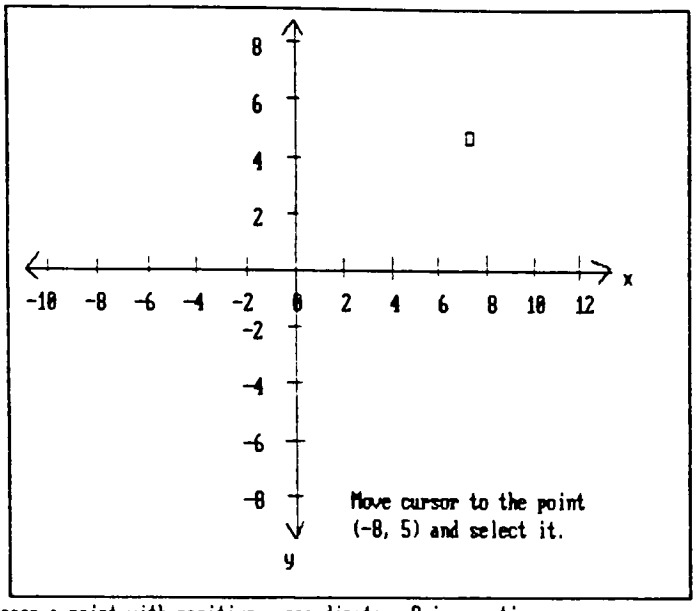
Right! That point is not in A and is in both B and C.
To continue, press any key.

AL01



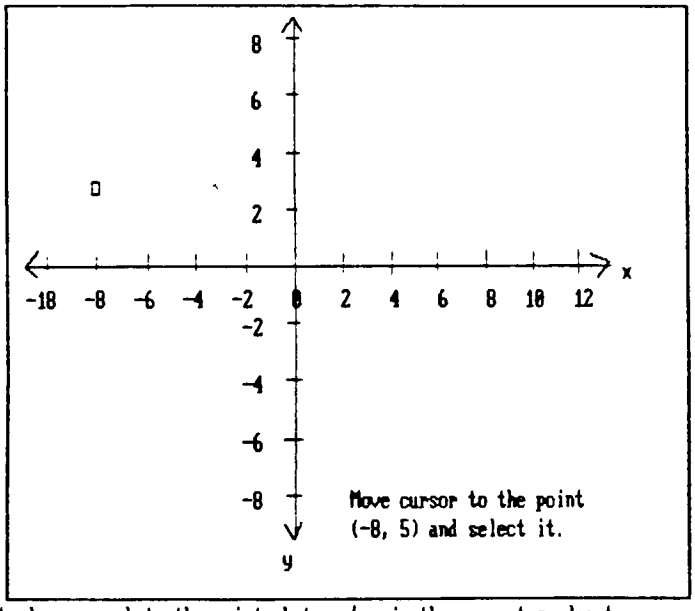
Waiting for answer selection

AL81



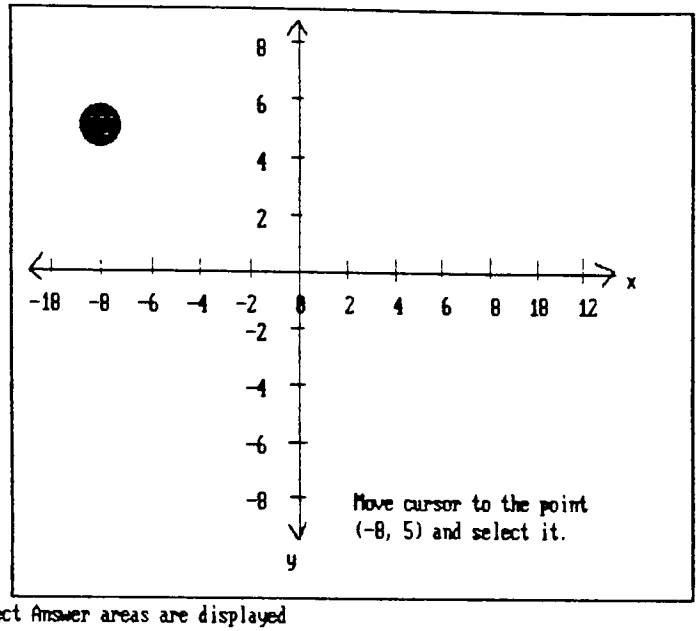
You've chosen a point with positive x coordinate. -8 is negative.
Try again. Select answer

AL81



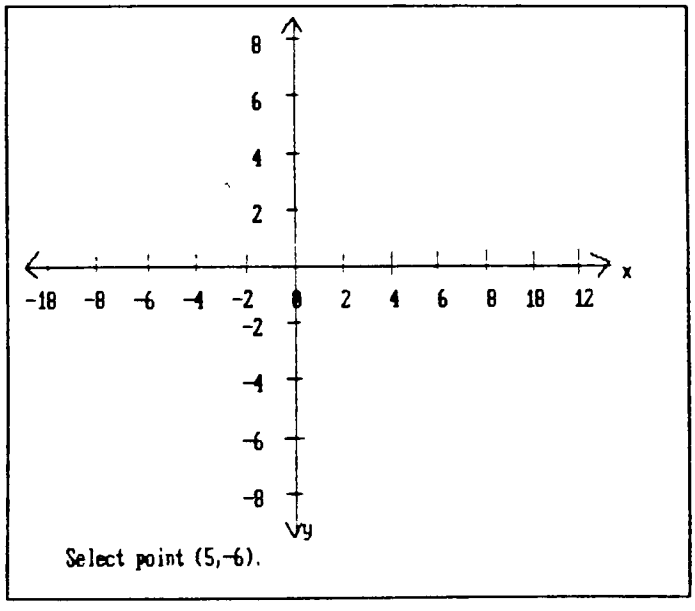
You're not close enough to the point, but you're in the correct quadrant.
For display of the correct answer(s), press any key.

AL81



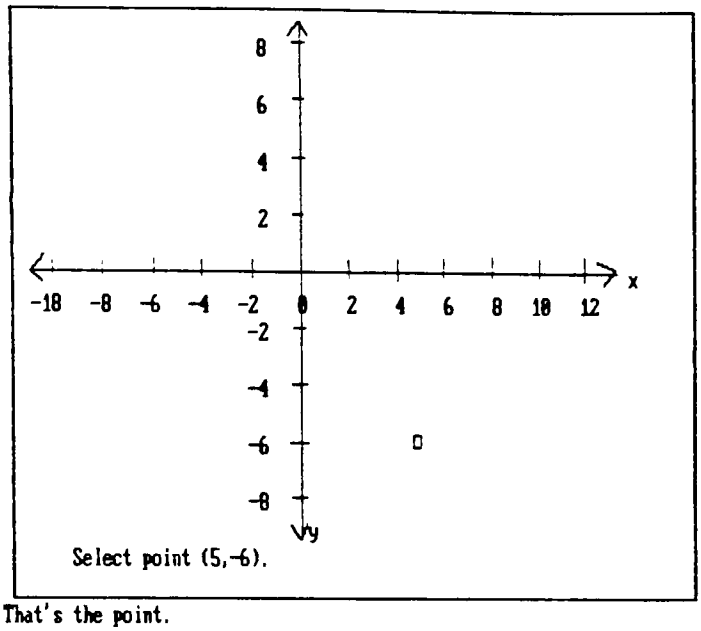
The Correct Answer areas are displayed
To continue, press any key.

AL84



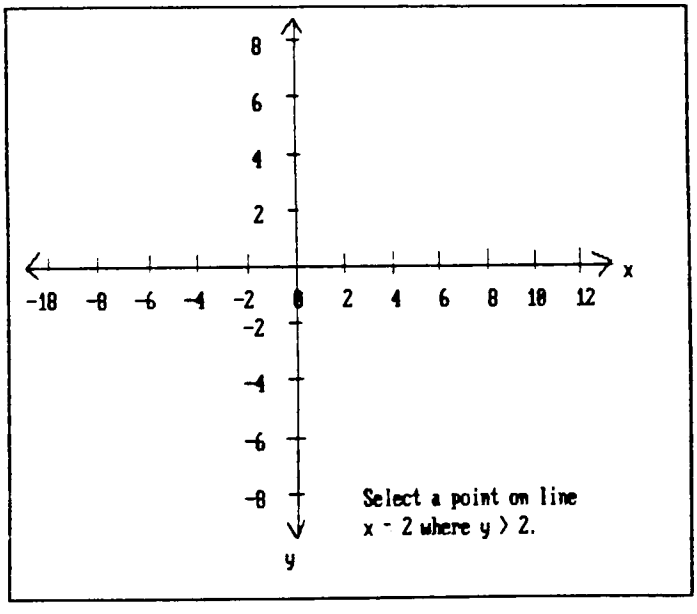
Waiting for answer selection

AL84



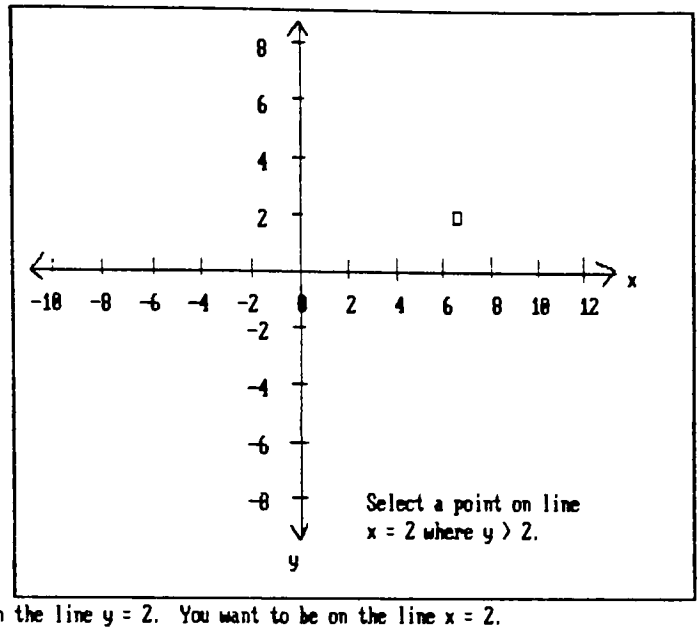
Correct! That's the point.
To continue, press any key.

AL82



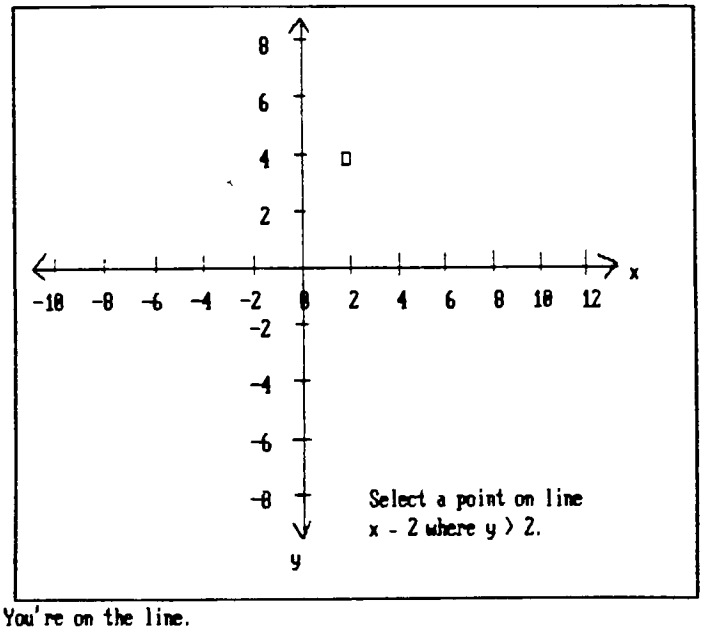
Waiting for answer selection

AL82



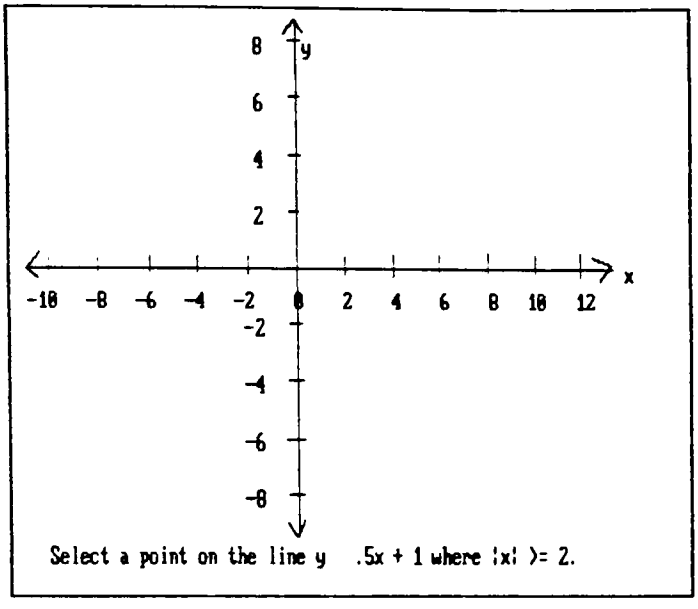
You're on the line y = 2. You want to be on the line x = 2.
Try again. Select answer

AL82



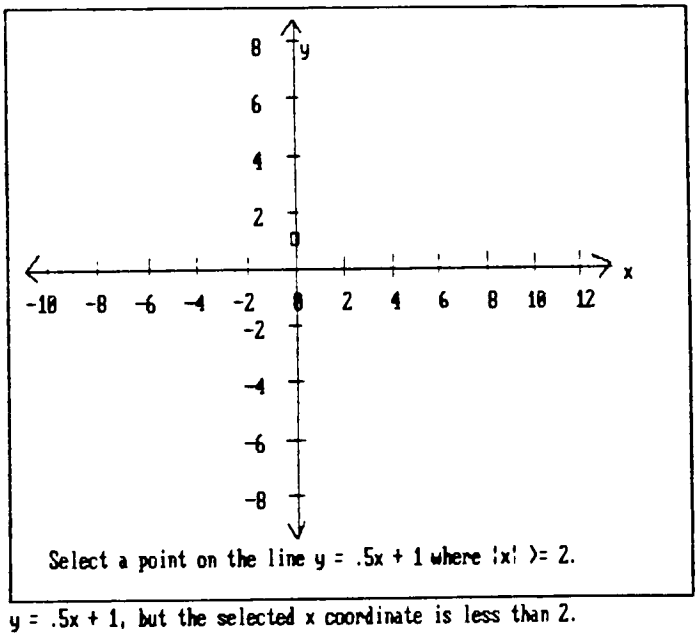
Correct! You're on the line.
To continue, press any key.

AL83



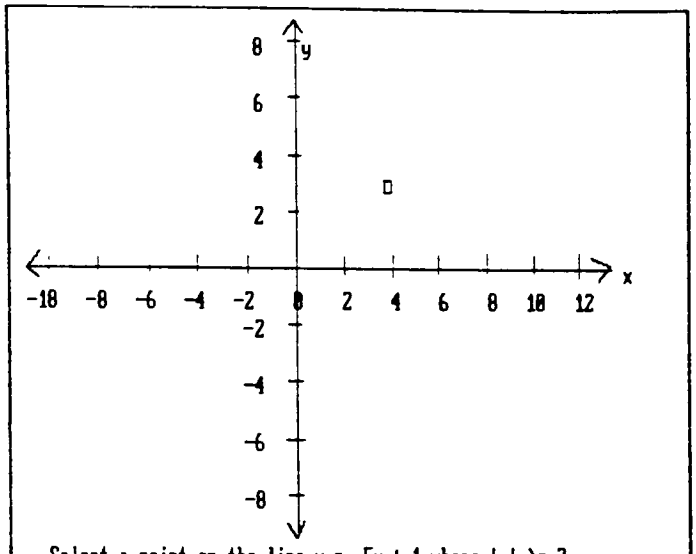
Waiting for answer selection

AL83



You're on y = .5x + 1, but the selected x coordinate is less than 2.
Try again. Select answer

AL03



Select a point on the line y = .5x + 1 where |x| >= 2.

Good! You've selected a correct point.
To continue, press any key.

You have successfully completed course ALGEBRA.

To continue, press any key.

Create frames using Microsoft PaintBrush (MAKEFRX.COM)

The MAKEFRX program allows for the conversion of a screen developed using Microsoft Paintbrush into the format used for frames in the Graphical Authoring System. It utilizes the FRIEZE utility by Microsoft which allows the displaying of a Paintbrush screen from another application program. In this case, the other application program is MAKEFRX. Procedurally, the author does the following steps to accomplish the conversion.

Use Paintbrush to create the screen display. If the Paintbrush screen has already been created, it is still necessary to enter and exit Paintbrush in order to load the FRIEZE utility into memory.

Type MAKEFRX<ret>

Press Shift-PrtSc to run the FRIEZE utility while in MAKEFRX.

Choose option L which loads a Paintbrush file and type the file specification of the Paintbrush file to be converted. This file will be displayed on the screen.

Press any key and respond to the filename prompt with the course frame ID. A file frame-id.FRX is created that can be used when authoring a course.

4. DESIGN AND IMPLEMENTATION ISSUES

This chapter discusses the significant design and implementation issues dealt with during the development of the Graphical Authoring System. The methods used by the system to save the author's specifications of frames, answers and the association of course components are explained, along with how this information is later utilized by the delivery of the course to the student. A detailed description of fill algorithm is also presented.

The Graphical Authoring System was developed on an Epson Equity II (IBM PC XT compatible) operating under MS-DOS with a Hercules Monochrome Graphics Adapter, 20 Megabyte hard disk and Microsoft Mouse (Bus Version). It was programmed with Borland's Turbo Pascal version 3.0 and makes use of the utilities provided with the Graphix and Database Toolboxes. Graphics hardware dependent values reside in constant include files which allow for adaptation to graphics hardware supported by the Borland Graphix Toolbox (Hercules Monochrome, IBM, and Zenith Graphic Adapters).

Answer Specification and Evaluation

The system allows for the specification of four types of answer areas. These areas are bounded portions of the screen that the author identifies as right or wrong answer selections and for which a response to be given to the student is specified. For the option frame (not implemented), the option choices would be areas defined in the same way and a next course step would be specified.

An implementation objective was to minimize the amount of disk space required to represent the answer areas while also keeping the time to evaluate a student's answer at a reasonable level. For two of the answer types this was trivial. For a circle, the center point and radius are saved. For a box, the upper left and lower right corner points are saved. For the fill and polygon area types, the specification of the areas and evaluation of whether a point is within the area is much more complex.

The fill option allows the author to specify the area within a fully bounded shape in the question frame as the answer area. After the specification of an answer, all points selected by the fill option are identified by saving a specific subset of the points. This subset of points is selected in the following manner.

The answer specification algorithm scans each horizontal line on the screen. For each line, the segments of the line that consist of points that were selected through the fill option are identified.

(See FillArray described in later section on fill algorithm.) The segments are contiguous pixels on the line. If the number of segments identified for the current horizontal line being scanned is different from the previous horizonal line scanned, the first endpoint of each segment is saved. Also, if the scan has proceeded for 50 lines without a change in the number of segments, the first endpoint of each segment is saved. Although this algorithm causes a slight delay for the author, the amount of disk space required to specify the area is minimized and the time required to evaluate the student's answer is reasonable.

Fill Answer Specification Illustration:

```
    Before                          After
..........bbbb...          ..........bbbb...
..........b..b...          ..........bFfb...
...bbbbb..b..b...          ...bbbbb..bffb...
...b...b..b..b...          ...bFfffb..bFfb...
..b....bbbb..b...          ..bfffffbbbbffb...
..b..........b...          ..bFfffffffffb...
...bbbbb....b.b..          ...bbbbbffffb.b..
........b..b.b.b.          ........bFfb.bFb.
........b..b.b.b.          ........bffb.bfb.
........bbb...b..          ........bbb...b..

b - border of shapes that were filled
f and F - points that fill the shapes
F - points that are saved as the Fill Answer Specification
. - points outside the shapes
```

When evaluating whether a student's answer is within a fill area, a neighborhood around the answer point is identified by doing the fill algorithm for 50 lines below the chosen point. If the fill does not extend 50 lines below the chosen point, the remainder is done above the point. This neighborhood will include one of the points in the fill area specification if the author specified a fill area that encompassed the student's selected answer point.

The polygon answer specification allows the author to specify the area within a polygon as an answer area by specifying the vertices of the polygon. The coordinates of these vertices are saved as the polygon area specification.

During the evaluation of whether a selected answer is within a polygon the horizontal line through the selected answer is checked for intersection with each of the sides of the polygon (the lines between each pair of consecutive vertices). The intersection is determined in two steps.

1. The range of the y coordinate values of the polygon side are indicated by the y coordinates of the vertices. If the y coordinate of the selected answer is within this range, an intersection exists.

2. If an intersection exists, the intersection is at x coordinate value, which is determined as follows.

Given (x1,y1) and (x2,y2) are the vertices, (x_ans,y_ans) is the selected answer and (x_int,y_ans) is the intersection, then

slope = (y2-y1)/(x2-x1)

b = y2 - (slope * x2)

x_int = (y_ans - b)/slope

Once all the intersections are determined, the x coordinate values of the intersections are ordered in ascending order. The selected answer is within the polygon if its x coordinate is between the first and second intersections or the third and fourth intersections or the fifth and sixth intersections, so on.

That is, given the ordered intersections of y = y_ans with the sides of the polygon are denoted as (x_int[i],y_ans) for i=1..n where x_int[i] <= x_int[j] when i < j,

if x_int[j] <= x_ans <= x_int[j+1] where j is odd

then (x_ans,y_ans) is within the polygon.

Illustrations of evaluation of point selections within a polygon:

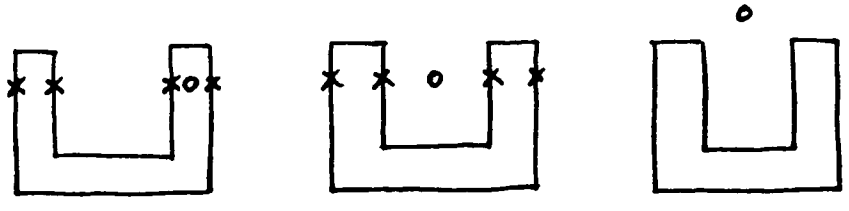


fig 1 fig 2 fig3

x marks intersection with y=y_ans
o marks point selection (x_ans,y_ans)

Fig 1. Answer is within polygon; answer is between 3rd and 4th intersection.
Fig 2. Answer is not inside polygon; answer is between 2nd and 3rd intersection.
Fig 3. Answer is not inside polygon; y=y_ans does not intersect the polygon.

Saving and retrieving course frames.

Borland's Graphix Toolbox provides routines for saving the screen display in a disk file and loading the screen display from a disk file. Since these routines do not compress the information, using one bit for each pixel, the saving of one screen requires over 32000 bytes (1 bit for each of 720x350 pixels) in the case of the Hercules Graphics Adapter. In order to reduce the amount of space required for saving course frames, routines to compress the information were written. The amount of space required for the saving of a frame will vary considerably depending on the distribution of the pixels illuminated on the screen. In the vast majority of cases the new SaveCourseFrame routine is expected to require considerably less space than the Borland SaveScreen.

For each horizontal line of the screen, the pixels are considered eight at a time (on byte boundaries). For consecutive groups of 8 pixels that are not illuminated, this information is stored in two bytes. The first byte has a value of 0 and the second byte is the number of consecutive groups. In the same fashion, for consecutive groups of 8 pixels that are illuminated, this information is stored in two bytes. The first byte has a value of 255 and the second byte is the number of consecutive groups. For all other groups of 8 pixels one byte is stored with one bit per pixel. Also, since the entire screen is not used for the course frame, no information is saved for groups of 8 pixels that are totally outside the course frame boundary on the screen.

Fill algorithm

The fill algorithm which is used to fill shapes when developing frames, identifying the fill pixels in answer specification, and identifying the answer neighborhood during answer evaluation is a slightly limited fill shape algorithm. Under certain circumstances the entire shape will not be filled. In these cases, the author can request the unfilled portion to be filled. Limiting the fill algorithm speeds it up while still being complete in a majority of the cases.

In general, a shape is filled by scanning horizontally within the shape. A pixel within the shape and next to the boundary is put in a queue as it is lit in order to later scan starting from the pixels immediately above and below it. A data structure named the FillArray which has one bit per pixel is used. The bit corresponding to a pixel is set in the FillArray if the pixel is lit as a result of a fill.

The algorithm starts at the chosen pixel. If the chosen pixel is lit, the algorithm is finished. If not finished, pixels to the right of the point are checked until a lit pixel is found. Each pixel along the way that was not lit is lit and its corresponding bit in the FillArray is set. The coordinates of the last pixel to be lit are put in a queue along with the direction the scan was moving. This same activity is done to the left of the chosen pixel.

For each pixel/direction in the queue (for as long as there's something in the queue), one of the following is done for both the pixel immediately below and above the pixel:

- o If the pixel's corresponding bit in the FillArray is set, nothing is done.
- o If the pixels's corresponding bit in the FillArray is not set and the pixel is not lit then pixels are scanned in the direction that came off queue until a lit pixel is found. The last unlit pixel and the current direction of the scan is put on the queue. Scanning is now done in the opposite direction and pixels are lit and their corresponding bits in the FillArray are set until a lit pixel is encountered. The last pixel to be lit is put in the queue along with the current scanning direction.
- o If the pixel's corresponding bit in the FillArray is not set and the pixel is lit then pixels are scanned in the opposite direction than the direction from the queue until one of two following conditions:

    the pixel is lit and its corresponding FillArray bit is set

    the pixel is not lit

    If the second condition is true and either the pixel below or above the found pixel has its FillArray bit set, then do the following

    Continue in the same direction until a lit pixel is encountered. The last unlit pixel and the current direction of the scan is put in the queue. Scanning is now done in the opposite direction and pixels are lit and their corresponding bits in the FillArray are set until a lit pixel is encountered. The last pixel that was lit along with the current scanning direction are put in the queue.

This algorithm is deficient in that it does not put each pixel that is lit in the queue, instead it only does endpoints. As a result much redundant checking is avoided but certain special cases are not fully filled. The following illustration demonstrates one of those cases.

```
          Before
.............................................
.bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb.
.b.........................................b.
..b.......bbbb....bbbbbbbbb.............b..
..b.......b..b....b.......b.............b...
..b......b....b...b.......b.............b...
..b......b....b...b.......b.............b...
..b......bbbbbb...bbbbbbbbb...........b....
..b.....................................b....
..bbbbbbbbbbbb.....................bbbbbb....
............bbbbbbbbbbbbbbbbbbbbb.........
.............................................

          After
.............................................
.bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb.
.bfffffffffffffffffffffffffffffffffffffffffb.
..bfffffffbbbb----bbbbbbbbbfffffffffffffb..
..bfffffffb..b----b.......bfffffffffffb...
..bfffffffb....b---b.......bfffffffCffffb...
..bfffffffb....b---b.......bfffffffffffb...
..bfffffffbbbbbb---bbbbbbbbbfffffffffffb....
..bfffffffffffffffffffffffffffffffffffb....
..bbbbbbbbbbbbbfffffffffffffffffffbbbbbb....
............bbbbbbbbbbbbbbbbbbbbb.........
.............................................
```

each letter represents a pixel:

. - unlit at start
b - pixel lit at start
f - pixel not lit at start, lit as result of fill
C - pixel not lit at start, pixel chosen within shape to fill, lit after fill
- - pixel not lit at start that would be lit with a complete Fill Shape algorithm

Stored Data

The data that is required to be saved for later retrieval by the system is maintained in two ways: through the Turbo Database Toolbox and in sequential flat files.

Turbo Database Toolbox maintains index files and data files through its B+ tree structure. The Graphical Authoring and Delivery System uses four index files and two data files.

STEPFILE.NDX - index file containing the steps that the courses consist of

STEPFILE.DAT additional information for question steps and lesson steps which is accessible through STEPFILE.NDX

COMPFILE.NDX - index file containing information on the existence and association of key components of the system.

ANOPFILE.NDX - index file containing the answers for questions and option choices for option step.

ANOPFILE.DAT - answer type and answer response information for answers; next step information for option choices.

PERFFILE.NDX - index file containing the current step in each course for each student.

The database design tried to minimize the number of open files required; to allow for quick identification of existence of certain component or component relationships; and to utilize the ordering of the index keys so that retrieval of data that is required in a specific order would be quick. For specifics of the formats of the index keys and data records see Appendix B.

Each frame display is saved in a file named frame-id.FRM consisting of records of 200 bytes containing compressed pixel information as described in the section titled *Saving and retrieving course frames* in this chapter.

Each question's answer area specifications are saved in a file named frame-id.ANS. Each record in the file consists of 6 integers. The first record of the file identifies the record displacement from the start of the file to each (up to 5) of the answer area specifications. See Appendix B for the specific layout of this file.

Keyboard Data Entry

The routine INPUTSTRING was developed to accept a data field from the keyboard that has certain attributes. The characteristics of the field that can be specified are:

- where to display the field on the screen
- length of the field
- type of field (alphanumeric, numeric, course step type)
- left justification
- auto entry
- input required
- upper case only

- embedded blanks allowed
- full field required
- exit allowed

A field can be terminated by the forward or back tab, enter, and the F10 function key when exit is allowed. The terminating key is returned from this routine to be used by multiple field inputting.

Overlaying

For programs requiring more then 64KB, Turbo Pascal provides overlaying capabilities. When using a hard disk, this overlaying does not have a noticeable negative impact.

5. EVALUATION

This chapter focuses on the restrictions, limitations and deficiencies of the Graphical Authoring System.

Frame Development

During frame development, the system acts as a drawing pad with the mouse functioning as a pencil. The ERASE option allows for the erasing of the drawing similar to an eraser. When drawing on the frame with the drawing options the system does not retain any information regarding objects (such as circles and rectangles). The frame is only recognized as lit pixels on a frame portion of the screen. UNDO and CLEAR are available to the author as a convenient means of backing out of some of the drawing activity. UNDO removes the effects of the last drawing option. The CLEAR option returns to a clear frame when adding a frame and removes all the effects of the current modify session when the frame is being modified. READ allows the author to retrieve a copy of a previously developed frame.

This method of developing a frame was chosen for its simplicity for both the course author and system implementor. The development of a frame is very similar to the activity the author would do when drawing a diagram in the traditional manner. More sophisticated functions such as cut and paste are available in Microsoft Paintbrush and the conversion program, MAKEFRX, is available to convert Paintbrush files to course frame files.

An object-oriented frame development in which object information is saved could provide the author with addtional capabilities to move, duplicate and size objects. Of particular concern would be how to save the description of a free form object (as accomplished with the DRAW option). It is expected that under the current environment using Turbo Graphix Toolbox that the display of the objects making up a frame would be slower than the display of the compressed pixel map in the current implementation. If this is true, a pixel map could be saved as well as the object descriptions. If the display is faster when redisplaying the objects, this would be a significant benefit in support of this method over the implemented method. In addition, saving the object descriptions would make possible the option of selecting an object as an answer area.

Composite Frames

Each frame is saved in a separate frame file. During the development of a course, the frames required may be very similar. A means of saving the similar information for use by similar frames would provide more efficient use of disk space. An attempt has already been made to minimize the space required to save individual frames by compressing information about the pixels that make up the frame. Any method of saving information that is similar across frames would make modification and deletion of frames more complex for the author to use. To allow for the specification of a "common frame" and the specification of overlaying text and the starting point of the text would be the most attractive alternative to compress information across frames. It would be the least extensive to implement and would minimize the complexity added to the author's activities.

Answer Area Specifications

It is the responsibility of the author to reverify the appropriateness of answer area specifications for a frame when a frame has been modified. The review of answer specifications is available through the DISPLAY suboption for the ANSWER option. An enhancement to the system to automatically display the current answer area specifications for a modified frame and ask for verification of their correctness from the author would help to avoid an oversight by the author to reevaluate answer specifications.

The specification of a fill area requires extra care by the author to recognize which pixels are being considered part of the fill area. When a student's answer selection is evaluated against a fill area, the lit pixels of the frame are used to identify a neighborhood around the answer selection. It is important that a pixel lit as a result of display a circle, box or polygon answer area during an answer specification does not constitute part of the boundary for a fill answer area. Procedurally, this concern can be avoided by selecting fill areas first when developing an answer specification.

If text is within a shape, it is considered in the same way as any other lit pixel and is not part of the fill area that fills the shape. The author will need to specify the text as part of the answer area by using the BOX, CIRCLE, or POLYGON option to encompass the text within the shape. The composite frame alternative discussed earlier, in which the text is saved separately, may be helpful in including labels within a shape when specifying only the FILL option.

Data item field lengths

At present the author is limited to 7 characters for a course ID and 4 characters for frame IDs and step IDs. It is suspected that a larger length for frame ID and step ID would be helpful in order to provide more meaningful names. Evaluation of the optimal length should be done. Since these fields are stored in the database, their lengths affect the size of the database. Since frames and answer specifications are saved in files using the frame ID as part of the file name, another method would have to be used to name the file if the frame ID exceeded 8 characters (DOS file name limit).

Multiple data field entry

For data input requiring more than one field, such as the QUESTION STEP ADD screen, using the enter key seems awkward for indicating the completion of entering all the fields. There is a tendency to press the enter key after a field entry which prematurely completes the entry of the entire multiple field screen. The enter key may be more appropriate for the completion of a field entry (like a tab) with a function key used to indicate completion of entry of all the fields.

Drawing Device

The Mouse is appropriate for the selection activities but it is difficult for the author to use for drawing when using the DRAW option during frame development. Another device such as a pen pad may be more appropriate to facilitate drawing frames by allowing the author to trace a hardcopy of a drawing.

Course Deletion

The deletion of a course is not described in the original specification. It is a needed function. It would be must appropriate as a separate execution module because of the extent and criticalness of its modifications.

Menu Hierarchy

The development of frames, the specification of answer areas for question frames, and the grouping of questions frames into question groups can be done independent of course assignment. In fact, frames and question groups can be used by multiple courses. For this reason, it may be appropriate to have the FRAME, ANSWER, and QUESTION GROUP options available also on the main menu. For the convenience of the author, it is legitimate to retain them on the course development menu.

Speed of graphical processing

The Graphical Authoring System would be enhanced by speeding up the following activities:

- saving answer area specification
- filling a shape
- displaying a frame on the screen
- saving a frame

Saving an answer area specification is time consuming when a fill area has been specified. This is due to trying to minimize both the space requirements for the answer specification and the time required to evaluate the student's answer. This delay in processing could be perceived as even longer than it is since no activity is occurring on the screen during the processing. Concentrating on making the code for this function more efficient would be appreciated by the course author.

The other activities take 1 to 3 seconds to process in the majority of the cases. Although not troublesome, speeding up these processes would certainly enhance the smooth running of the system. There is a noticeable delay in the display of a frame when another record is read from the frame file. The record size of the frame file should be optimized to balance the tradeoffs between file size and processing speed.

All the graphical processing utilizes Borland Graphix Toolbox routines. Replacing these routines to eliminate any unneeded overhead may lead to some gains. In the case of filling a shape, directing the hardware directly to do the fill, if the capability exists, would speed the fill display. For answer

specifications of a fill area, it would be necessary to save a "snapshot" of the frame before the fill so that it could be exclusive-OR'd with the resulting frame in order to identify the pixels that were lit as a result of the fill.

Database

A large portion of the stored data is course dependent. An alternative to the current database organization would be one that stored course dependent data in individual files for each of the courses. The system would create the course file(s) when the new course is added. It would not be necessary to repeat the course ID in these files since it would be apparent from the file(s) being used. Access, modification and deletion of course information would be simplified.

Detecting that the printer is offline

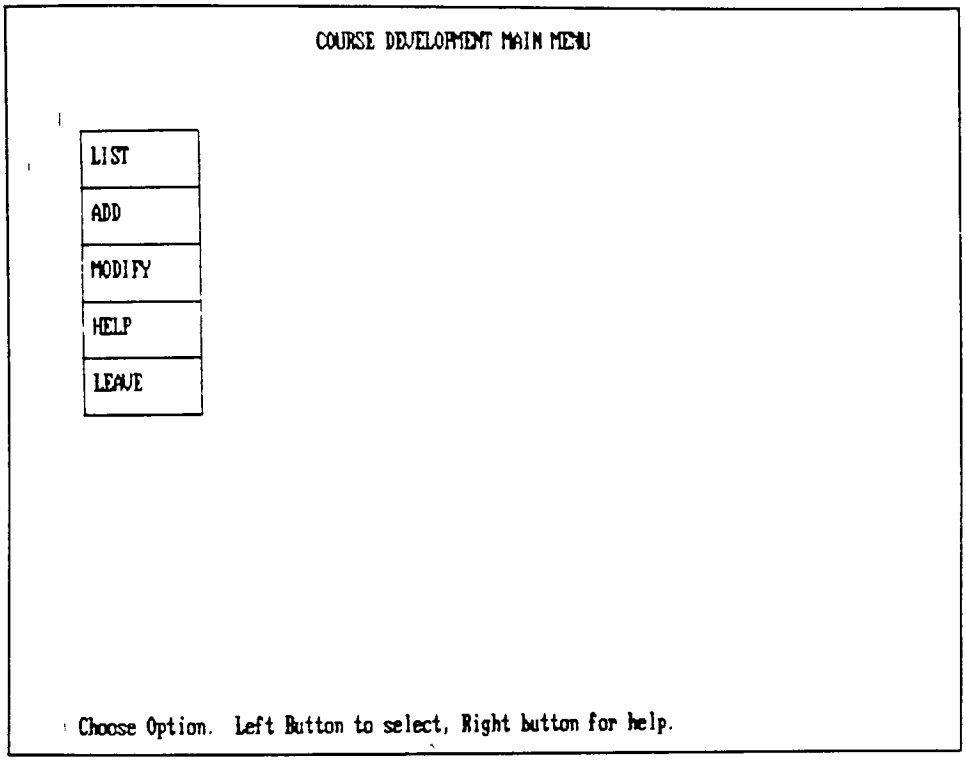
Currently, the authoring module is unable to identify when the printer is not online. If a print activity is requested when the printer is not online, a fatal error occurs.

Conversion of Paintbrush file to course frame (MAKEFRX)

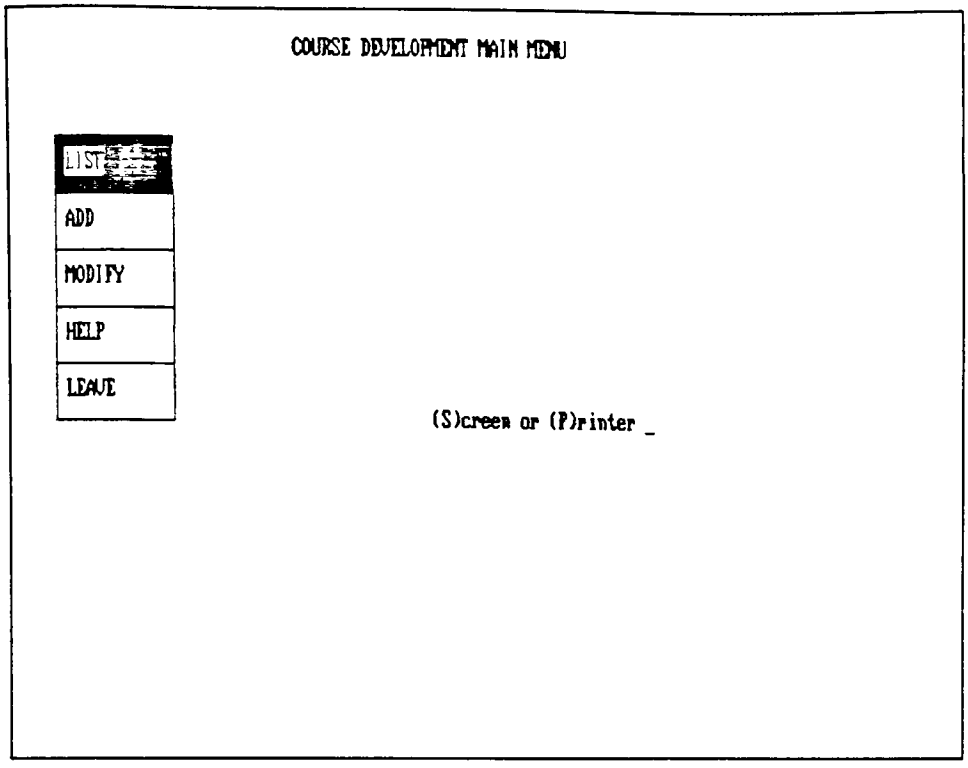
When running MAKEFRX to convert a Microsoft Paintbrush screen to course frame format, the menu that is displayed when initiating the Microsoft FRIEZE program is not readable. Since it is known that the load option is desired, the author can type L and enter the file name of the file to convert and the conversion is accomplished.

APPENDIX A
SCREENS AND HELP FILES

This appendix contains the screen, including the help screens, that can be seen by the author during an authoring session. This is to serve as a reference manual of the Graphical Authoring and Delivery Sytem's capabilities. The ---------- in the help files denotes the end of a screen.



Initial Authoring Main Menu



Main Menu with LIST selected

filename: MLIST.HLP

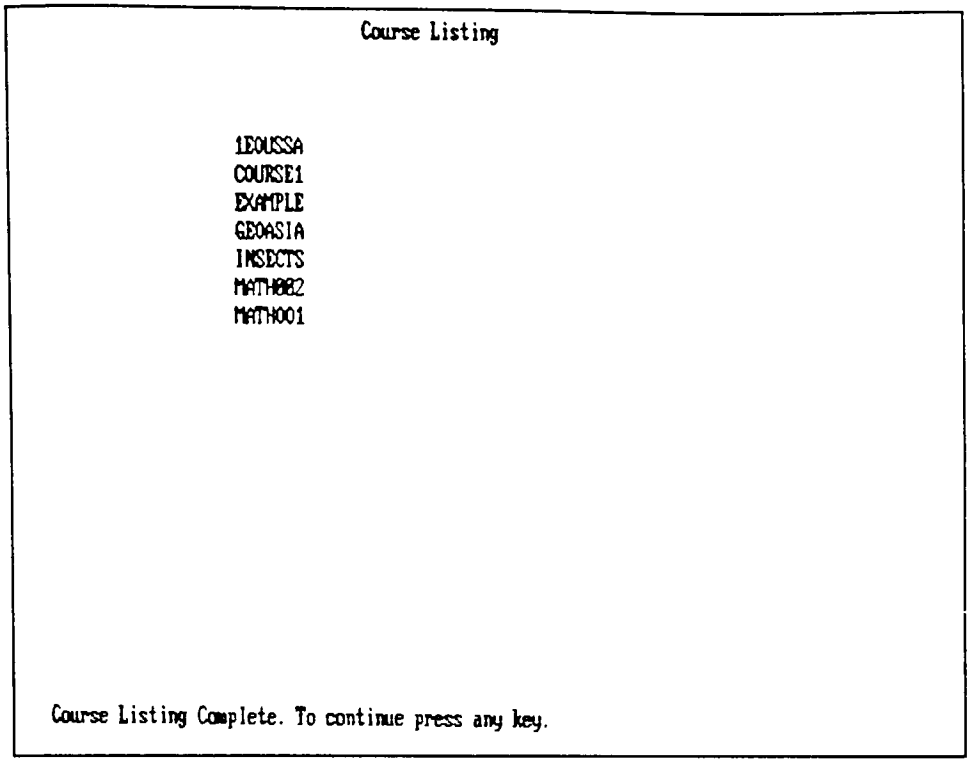
Main Menu
LIST option

Displays or prints a list of the courses that exist in the database.

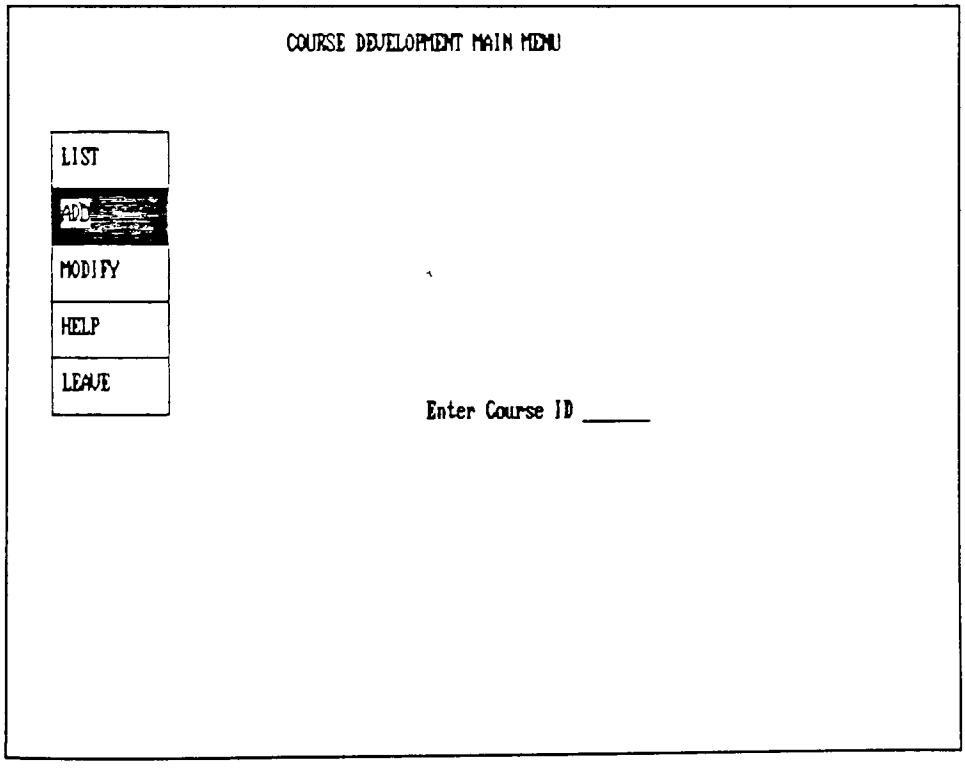
Respond to the 'Screen or Printer' prompt with an S or P followed by enter key.

If S is selected, the listing will go to the screen displaying one full screen at a time.

If P is selected, the listing will go to the printer. This capability is not currently available.



Resulting screen of Main Menu LIST



Main Menu with ADD option selected

filename: MADD.HLP

Main Menu
ADD option

This option adds a course to the course database and leads to the course development menu.

Respond to the 'Course ID' prompt with the ID of the new course. If the course already exists an error message is displayed. Use the Modify option on the Main Menu to make changes and additions to a course that already exists.

The Main Menu with MODIFY selected appears with the same prompt as the ADD option. Both ADD and MODIFY lead to the Course Development Menu. See subsequent presentation.

filename: MMODIFY.HLP

Main Menu
MODIFY option

This option leads to the course development menu for continued development of an existing course.

Respond to the 'Course ID' prompt with the ID of an existing course. If the course does not exist, an error message is displayed. Use the Add option on the Main Menu to start the development of a new course.

Selection of the HELP option, causes the display of the following help screens.

filename: MHELP.HLP

Graphical Authoring System
General Instruction

Two input devices are used in authoring a course: the keyboard and the mouse. The information that the system requires is identified by one or more of the following:

- directive information on the last two lines of the screen;
- data field prompts;
- the existence of mouse arrow cursor.

MOUSE INPUT:

A majority of the authoring is done using the mouse. The mouse is used to choose options and to identify points on the screen when developing course frames (lesson, question & option frames) and specifying answer responses to questions. The existence of an arrow on the screen (the arrow moves as you move the mouse) and a message at the bottom of the of the screen that directs you to 'select', 'choose' or 'point to' indicates that the system is awaiting a mouse input.

FIELD INPUT:

When entering data in a field, the data is entered through the keyboard and the following special keys have the following functions.

left arrow: to go left one character without deleting a character.
right arrow: to go right one character if a character exists to the right of the cursor.

---

tab: to finish the entry of the field and go to the next field if there is more than on field.
backtab: to finish the entry of the field and go to the previous field if there is one.
enter: to finish the entry all all fields on the screen
delete backarrow: to delete character to left of cursor.
DEL key: to delete the character under the cursor.
F10: to cancel the current activity (this key is not always allowed).

Note that the tab and backtab act the same as an enter when only one field is being entered.

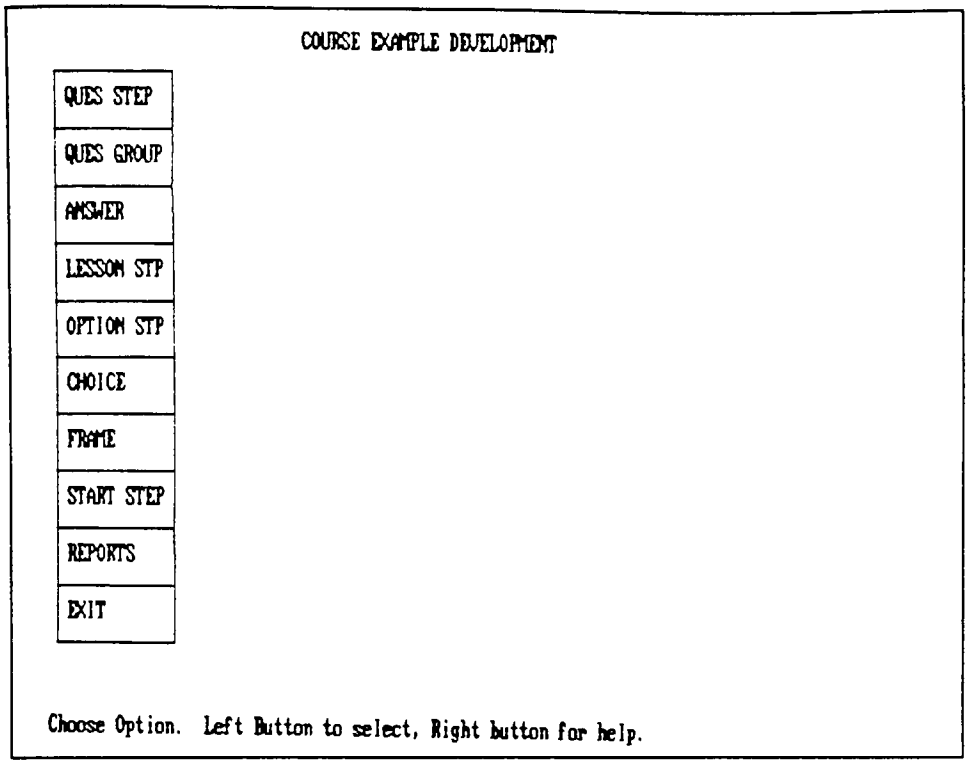
When errors are made during entering data, a descriptive message is displayed on the last line of the screen.

Main Menu LEAVE option

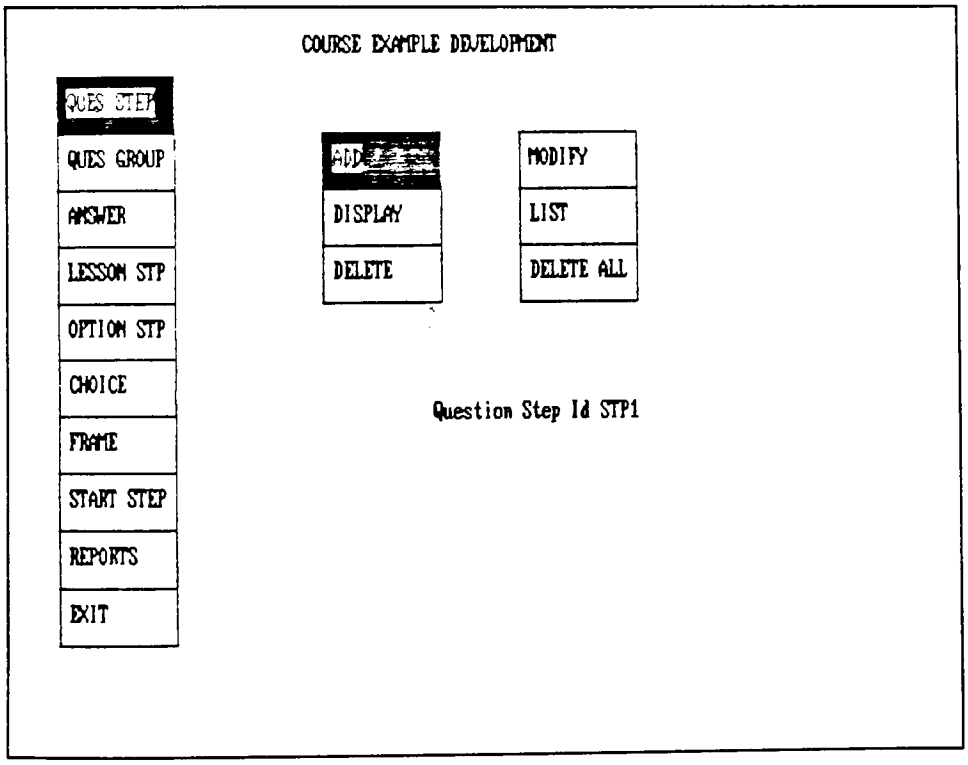
filename: MLEAVE.HLP

Main Menu
LEAVE option

Select this option if you wish to finish your authoring session.



Course Development Menu



Course Development Menu with QUEST STEP option selected

filename: MQUESSTP.HLP

Course Development
QUEST STP option

This option allows you to describe a question step for the course you are currently developing.

The attributes of a question step that can be specified are:

- The frame id of the question to be asked during this step of the course.
- The number of tries the student will have at answering the question correctly.
- The step that will be the next in the course if the student answers the question correctly within the number of tries specified.
- The step that will be the next in the course if the student is not able to answer the answer correctly within in the the number of tries specified.
- The other fields marked are for future use.

When entering the success and failure steps, the step has two parts to it - the step type and step id.

The step type can be Q or S or F.

Q is entered if the next step is a question step;
S is entered if the course has met successful completion;
F if entered if the course has met unsuccessful completion.

When a step type of Q is entered, the specific step id of the question step is entered in the field to the right of the step type. The step types of S and F do not require the entry of a step id.

(Future phases will add the capability to associate a group of questions to a question step and to assign performance criteria for successful answering of questions from the group.)

The Menu options for question step are:

ADD — Enter the step id of a new question step for the course being developed and press enter. An error will be displayed if the step already exists. On the subsequent screen, enter the FRAME ID, MAX TRIES, SUCCESS STEP, and FAILURE STEP as described above. Move from one field to the next using the forward & back tabs. When you have entered all the values you wish to identify at this time, press the enter key. If you wish to discontinue (abort) this option at any point, press F10.

MODIFY — Enter the step id of an existent question step for the course being developed and press enter. An error will be displayed if the question step does not exist. On the subsequent screen, update the FRAME ID, MAX TRIES,

SUCCESS STEP and FAILURE STEP fields to desired values. Move from one field to the next using the forward & back tabs. When you have made all your desired updates, press the enter key. If you wish to discontinue (abort) this option at any point, press F10.

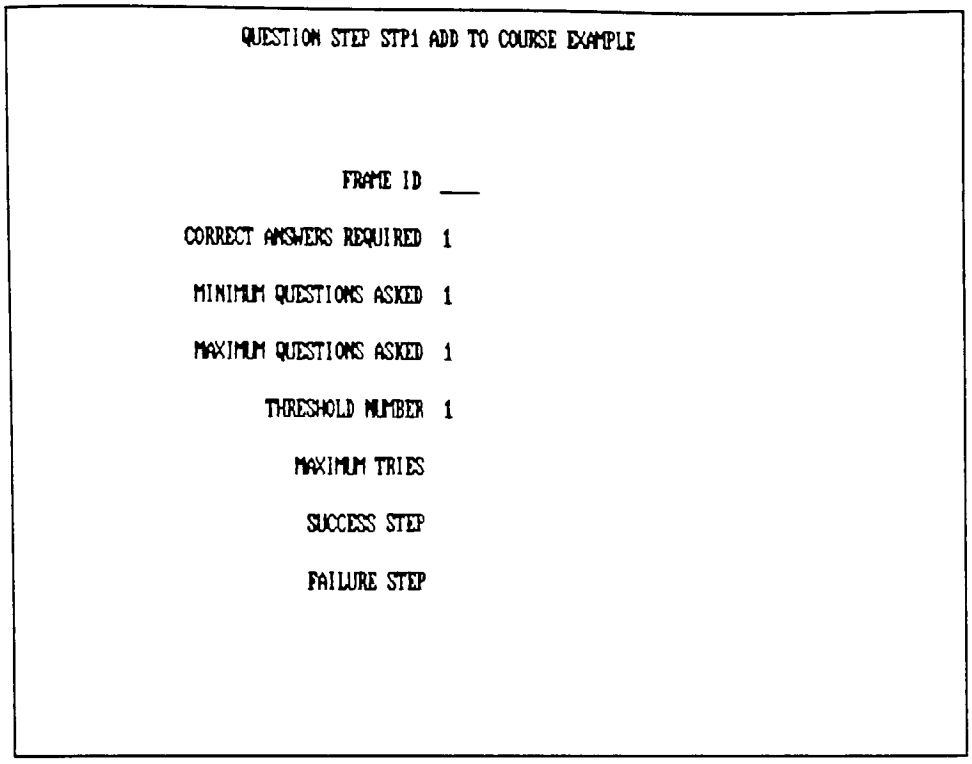
DELETE — Enter the step id of an question step that is currently specified for the course being developed. An error message will be displayed if the question step does not exist. If you wish to abort this option, press F10. After the question id is entered, you are asked to confirm that you wish to delete the question step. Respond with Y for yes and N for no.

DISPLAY — Enter the step id of an existent question step for the course being developed. An error message is displayed if the question step does not exist. The subsequent screen will display the current values of the question step attributes.

LIST — Unavailable

DEL ALL — Unavailable

The following is the screen for the ADD suboption of QUES STEP. The screens for the MODIFY and DISPLAY suboptions are the same format.



ADD suboption of QUES STEP option

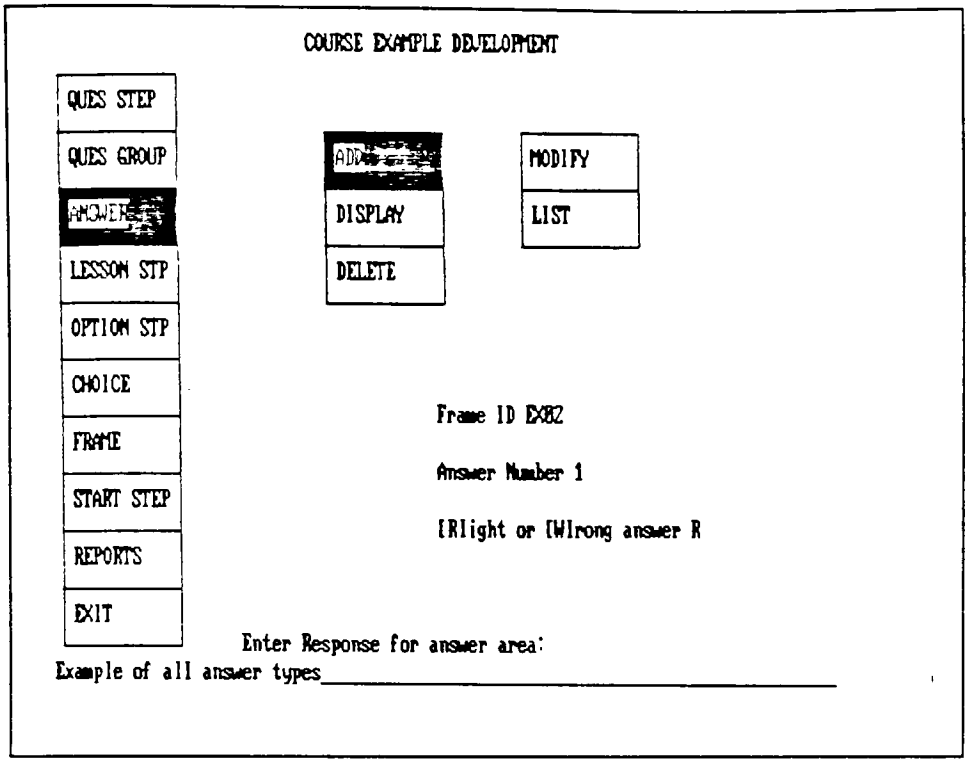
QUES GROUP option not available

filename: MQUESGRP.HLP

Course Development
QUES GRP option

This option is not available.

In the future it will allow you to assign frames (questions) to a question group.



Course Development Menu with ANSWER option selected

filename: MANSWER.HLP

Course Development
ANSWER option

This option allows you to specify answers for question frames.
For any of its options, described below, press F10 to abort.

ADD Enter the question frame id & answer number you wish to add. An error is displayed if the answer already exists. Valid answer numbers are 0 through 5. Use answer 0 to specify a default response. The system prompts you for the answer type and response. Respond to type with R for right or W for wrong. Reply to response prompt with the text you wish to display to the student when choosing a point within the area you are about to specify. In the case of answer number 0 it is the response given if the student chooses a point not within any of the specified areas. When adding answer number 1 through 5, a subsequent screen is displayed that allows you to specify the answer area(s). See the help screens for those options for further information.

MODIFY Enter the question frame id and answer number you wish to modify. An error is displayed if the answer does not already exist. You are prompted for the new answer type and response.

Respond to them as described in ADD option. The answer area specification can not be changed. You would have to do a DELETE and then ADD in order to change an answer area specification. (NOT AVAILABLE)

DELETE Enter the question frame id and answer number that you wish to delete. An error message is displayed if the answer does not exist. Respond to the message that confirms that you want to delete the answer with Y for yes and N for no.

DISPLAY Enter the question frame id and answer number of an existent answer that you wish to see. An error message is displayed if the answer does not exist. The subsequent screen displays the answer areas for your observation. Press any key to continue. (The answer area can not be modified from this option)

LIST Not Available

DEL ALL Not Available

LESSON STP option not currently available

filename: MLSNSTP.HLP

Course Development
LESSON STP option

This option is currently unavailable.

In the future this option will allow for a lesson step in the sequence of course steps.

OPTION STEP option not currently available

filename: MOPTSTP.HLP

Course Development
OPTION STP option

This option is not currently available.

In the future, this option would allow for the specification of an option step in the sequence of course steps.

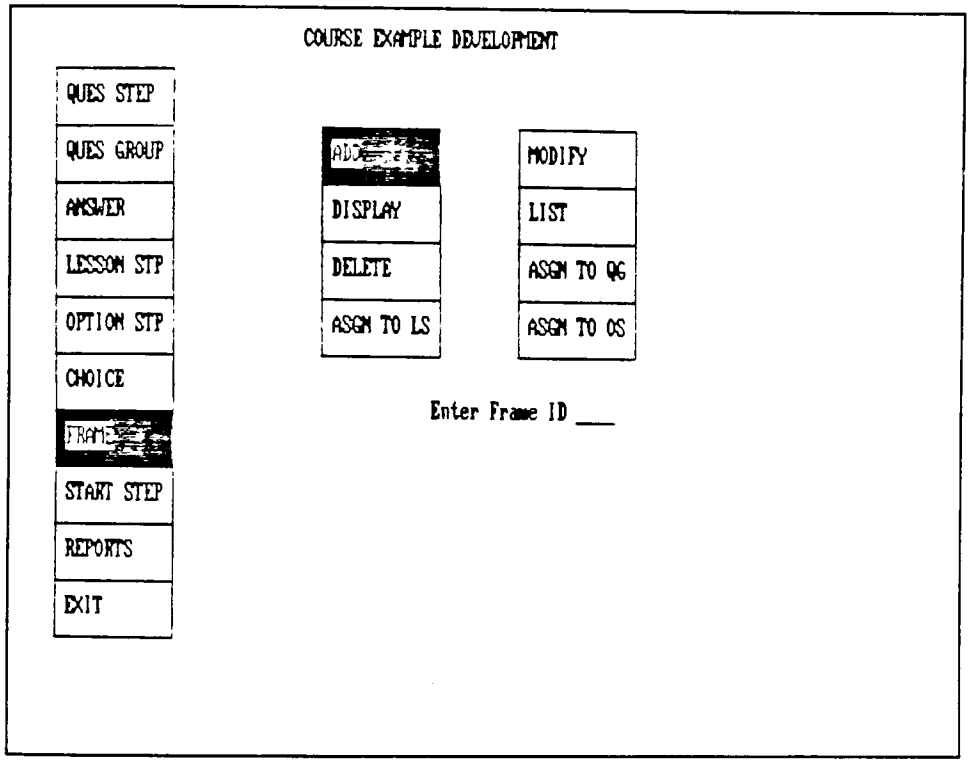
CHOICE option not currently available

filename: MCHOICE.HLP

Course Development
CHOICE option

This option is not currently available.

In the future, this option will allow for the specification of option choice areas for choosing an option on the screen.



Course Development Menu with FRAME option selected

filename: MFRAME.HLP

Course Development
FRAME option

This option allows you to maintain the frames (screen displays). For any of its options described below, press F10 to abort.

ADD Enter the frame id of the frame you wish to add. An error

is displayed if the frame already exists. The subsequent screen provides a list of options to be used to "draw" the frame. Help is available for each of the options.

MODIFY Enter the frame id of the frame you wish to modify. An error is displayed if the frame does not already exist. The subsequent screen provides a list of options to be used to "draw" the frame. Help is available for each of the options.

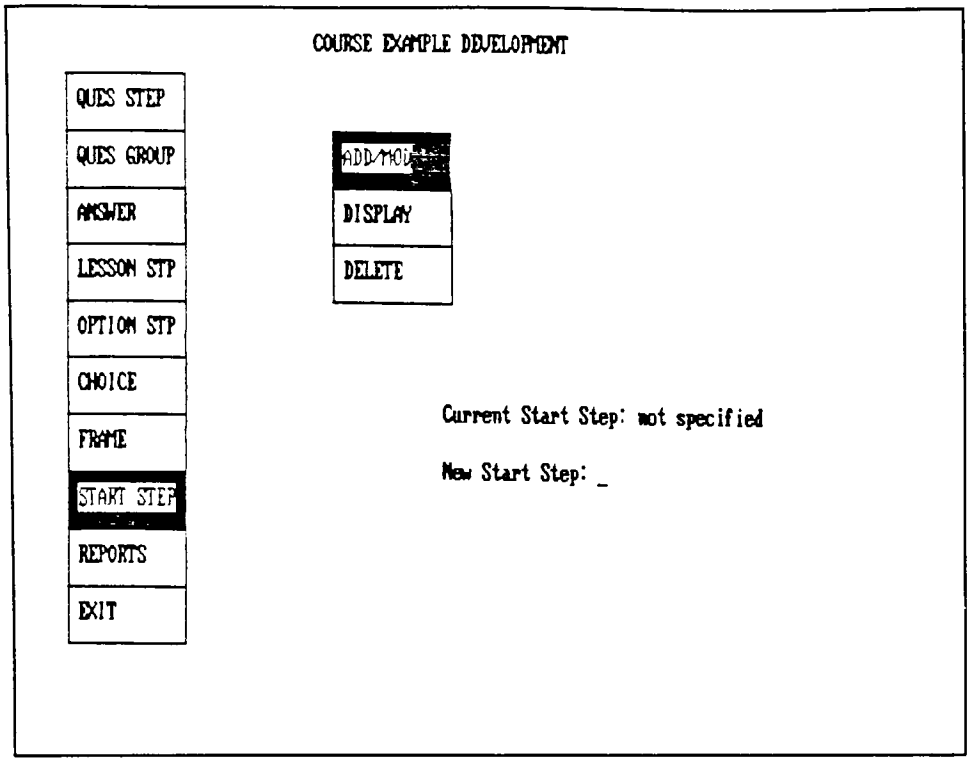
DELETE Enter the frame id of an existent frame that you wish to delete. An error message is displayed if the frame does not exist. Respond to the message that confirms that you want to delete the frame with Y for yes and N for no.

DISPLAY Enter the frame id of an existent frame that you wish to see. An error message is displayed if the frame does not exist. The subsequent screen displays the frame for your observation. Press any key to continue. (The frame can not be modified from this option)

LIST Not Available

DEL ALL Not Available

ADD and MODIFY suboptions of the FRAME option lead to a Frame Development screen. See subsequent descriptions of Frame Development options.



Course Development Menu with START STEP option selected

filename: MSTRTSTP.HLP

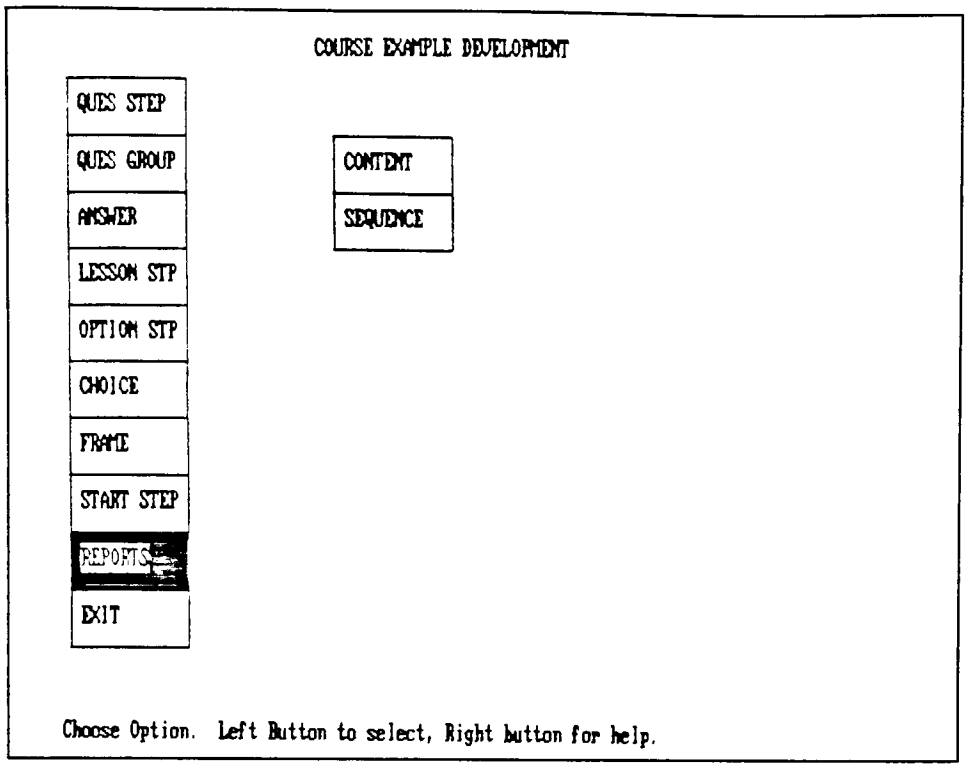
Course Development
START STP option

This step allows you to maintain the first step of the course being developed. To abort an option, press F10.

ADD/MOD The current start step of the course is displayed. Enter the step id of the new starting step. This step id is in two parts - the step type and the step id. For the step type press Q for question step (this is the only step type currently available.)
For the step id (2nd field), enter the step id of the desired question step.

DISPLAY Displays the start step of the course being developed.

DELETE Deletes the start step specification for the course being developed. Respond to the confirmation question with Y for yes and N for no.



Course Development Menu with REPORTS option selected

filename: MREPORTS.HLP

Course Development
REPORTS option

This option allows you to request the printing of reports that aid you in determining the completeness and validity of the course you are authoring. The two reports are the Sequence Report and Content Report. Select the appropriate option box in order to produce the report.

The CONTENT REPORT contains:

- For each question step in the course, the performance parameters and the frame ID assigned to the step.
- For each frame assigned to a step in the course, the steps it is assigned to, and flag indicating if the frame is not yet developed.

The SEQUENCE REPORT contains:

- The start step id
- For each step in the course, the steps subsequent to it and a flag indicating if the step is reachable through the current sequence.

Examples of Content and Sequence Reports

*** CONTENT REPORT ***

Course ID: EXAMPLE

*** QUESTION STEPS ***

Question Step ID	Frame ID	Max# Tries	Min# Asked	Max# Asked	Required Correct	Last Asked
STP1	EX01	2	1	1	1	1
STP2	EX02	1	1	1	1	1

** FRAMES ***

FRAME	Undef	USED BY STEP:
EX01		Question Step STP1
EX02		Question Step STP2

END OF REPORT

*** SEQUENCE REPORT ***

Course ID: EXAMPLE

Start Step: Question Step STP1

Undefined Steps

QUESTION STEPS

Step Number	Next On Success:	Next On Failure:	Unreachable
STP1	Question Step STP2	Failure End	
STP2	Success End	Failure End	

END OF REPORT

Course Development EXIT option

filename: MEXIT.HLP

Course Development Menu
EXIT option

Select this option in order to return to the main menu.

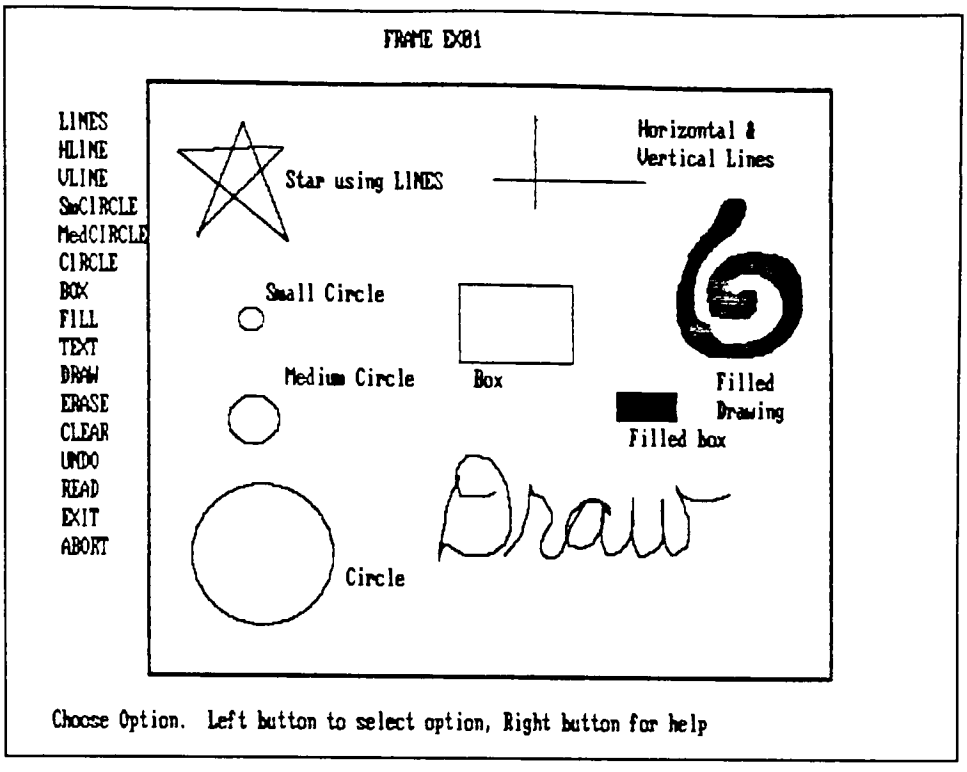
This is used when you
- wish to stop you authoring session
- wish to change the course for which you are authoring

Help screen for suboptions of Course Development Menu options

filename: MNHOPT.HLP

Course Development

Help for this option is available under the option choice one level up.



Frame Development screen

filename: FLINES.HLP

Frame Development
LINES option

This option allows you to draw connected straight lines.

1. Position the arrow at the starting point with the mouse.
   or
   Press the right button of the mouse to abort this option.
2. Press the left button of the mouse and a + sign marks this starting point.
3. Position the arrow at the other endpoint of the line.
4. Press the left button of the mouse and a line is drawn between the first point and the second point.
5. Position the cursor on the next endpoint.
   or
   Press the right button to complete this option.
6. Press the left button and a line will drawn between the second endpoint of the previous drawn line and the currently selected point. Go back to step 5.

filename: FHLINE.HLP

Frame Development
HLINE option

This option allows you to draw a horizontal line.

1. Position the arrow with the mouse on the first endpoint of the line
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and a + sign marks the the chosen endpoint.

3. Position the arrow at the second endpoint of the line. This is used solely to determine horizontal coordinate of the second endpoint.

4. Press the left button of the mouse and the horizontal line is drawn on the frame. (The second endpoint is forced to be at the same vertical coordinate as the first point in order to produce a horizontal line.)

filename: FVLINE.HLP

Frame Development
VLINE option

This option allows you to draw a vertical line.

1. Position the arrow with the mouse on the first endpoint of the line
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and a + sign marks the the chosen endpoint.

3. Position the arrow at the second endpoint of the line. This is used solely to determine the height of the line.

4. Press the left button of the mouse and the vertical line is drawn on the frame. (The second endpoint is forced to be at the same horizontal coordinate as the first point in order to produce a vertical line.)

filename: FSmCIRCL.HLP

Frame Development
SmCIRCLE option

This option allows you to draw a circle with a perdefined "small" radius.

1. Position the arrow at the center of the circle with the mouse.
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and the circle is drawn on the frame.

filename: FMedCIRC.HLP

Frame Development
MedCIRCLE option

This option allows you to draw a circle with a predefined "medium" radius.

1. Position the arrow at the center of the circle with the mouse.
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and the circle is drawn on the frame.

filename: FCIRCLE.HLP

Frame Development
CIRCLE option

This option allows you to draw a circle.

1. Position the arrow at the center of the circle with the mouse.
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and a + sign marks the the chosen center.

3. Position the arrow on a point on the circumference of the circle.

4. Press the left button of the mouse and the circle is drawn on the frame.

filename: FBOX.HLP

Frame Development
BOX option

This option allows you to draw a box (rectangle).

1. Position the arrow with the mouse on the top left corner of the box
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and a + sign marks the the chosen corner.

3. Position the arrow on the bottom right corner of the box.

4. Press the left button of the mouse and the box is drawn on the frame.

filename: FFILL.HLP

Frame Development
FILL option

This option allows you to fill a bounded shape.

1. Position the arrow at a point inside the shape to be filled with the mouse.
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and the shape is filled.

Be sure the shape is fully bounded. That is, there are no openings in the boundary through which the fill can "escape".

If the fill illuminates more of the frame than is desired, then use the UNDO option immediately. Do not select any other options prior to selecting UNDO.

The amount of "filling" you do on the frame, will limit the amount of "fill" answer specification that can be done. See ANSWER option on course development menu.

filename: FTEXT.HLP

Frame Development
TEXT option

This option allows you to enter text on the frame.

1. Position the arrow at the starting point of the text and press the left button. The cursor will move to the closest character position. This starting point also identifies the left and top margin of the text.
   or
   Press the right button of the mouse to abort the option.

2. Type the text as desired using the following keys as desired to move the position of the cursor.

ForTab	Go to the beginning of the next line using the left margin as specified by the starting point.
BackTab	Go to beginning of the previous line.
UpArr	Go to character position above current position.
DownArr	Go to character position under current position.
LeftArr	Go to character position to the left of current position.
RightArr	Go to character position to the right of current position.
Delete	Go to character position to the left deleting the character to the left.

3. When finished typing the text, press the enter key.

filename: FDRAW.HLP

Frame Development
DRAW option

This option allows you to 'draw' with the mouse.

1. Position the arrow to the point where you wish to start drawing with the mouse.
   or
   Press the right button of the mouse to abort the option.

2. While you depress the left button of the mouse and move the mouse a line follows it. You may stop drawing by releasing the left button and resume drawing by pressing the left button again.

3. When finished drawing, press the right button.

Hint: If you are going to do a lot of drawing, do it in small portions by exiting and rentering the DRAW option, this way an UNDO will only erase the last portion when a mistake is made. (see UNDO help)

filename: FERASE.HLP

Frame Development
ERASE option

This option allows you to erase portions of the frame display.

1. Position the arrow at the point where you want to start erasing.
   or
   Press the right button to abort the option.

2. While you press the left button and slowly move the mouse, the display will be erased under the 'eraser'. The eraser is a rectangle about the size of a text character above the arrow. When you stopping pressing the left button the erasing stops and you can resume erasing by pressing the left button again.

3. Press the right button of the mouse when you are done erasing.

filename: FCLEAR.HLP

Frame Development
CLEAR option

This option allow you to restart the frame development session.
If you are doing a frame add, the frame is cleared.
If you are doing a frame modify, the original frame is displayed.

1. A prompt will be displayed asking if you really want to clear.
2. Respond with Y if you do want to clear
   or with N if you want to leave the screen as is

filename: FUNDO.HLP

Frame Development
UNDO option

This option to undo the affects of the last drawing option.
(This includes LINES, SmCircle, MedCIRCLE, CIRCLE, BOX, FILL, TEXT, DRAW, ERASE and CLEAR)

Restriction: If Help screen was requested since the last drawing option, UNDO will have no effect.

filename: FREAD.HLP

Frame Development
READ option

This option allows you to display another frame created through Frame Development or an external frame created through the MAKEFRX facility. The current display will be erased.

1. A prompt will be displayed asking whether you wish to read a frame.

2. Respond with Y if you wish to specify a frame to display
   or with N if you wish to abort this option

3. A prompt will be displayed asking if the frame is an internal or external frame.

4. Respond with F for a internal frame (created with Frame Development)
   or with X for a external frame (created with MAKEFRX Facility)

5. A prompt will request the frame name

---

6. Respond with the frame id.

7. If the frame file exists, it will be read and displayed. If not an error will be displayed and the display will stay the same.

filename: FEXIT.HLP

Frame Development
EXIT option

This option allows you to exit the Frame Development and save the currently displayed frame.

1. A prompt will be displayed asking if you really want to leave the Frame Development and save the frame as displayed.

2. Respond with Y is you want to exit
   or with N if you want to abort this option.

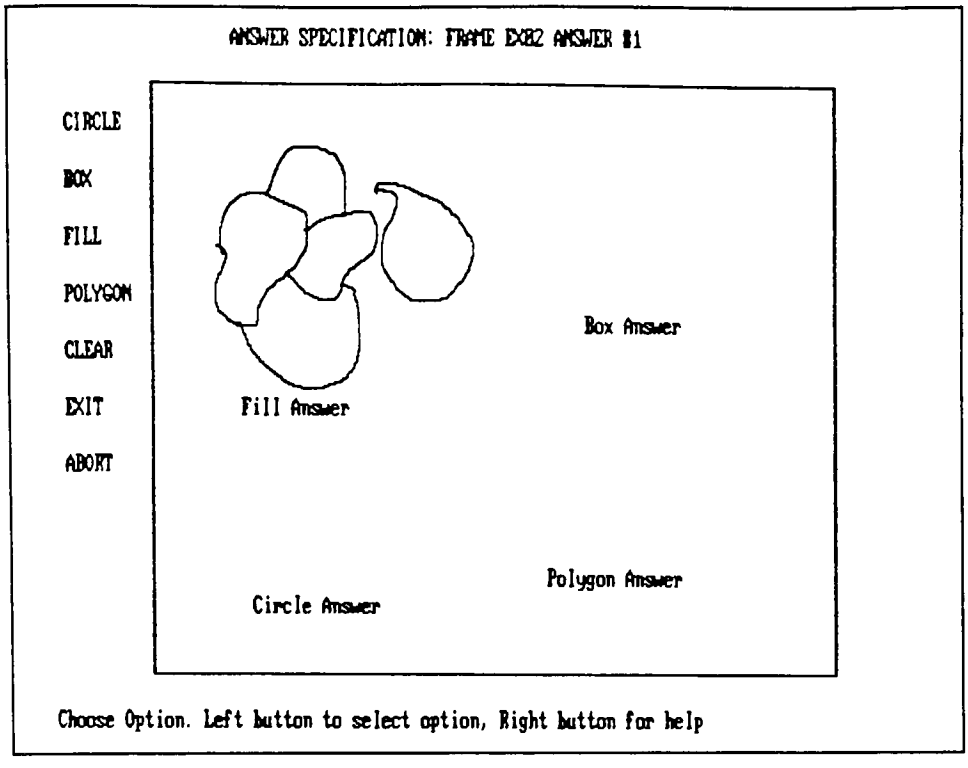
filename: FABORT.HLP

Frame Development
ABORT option

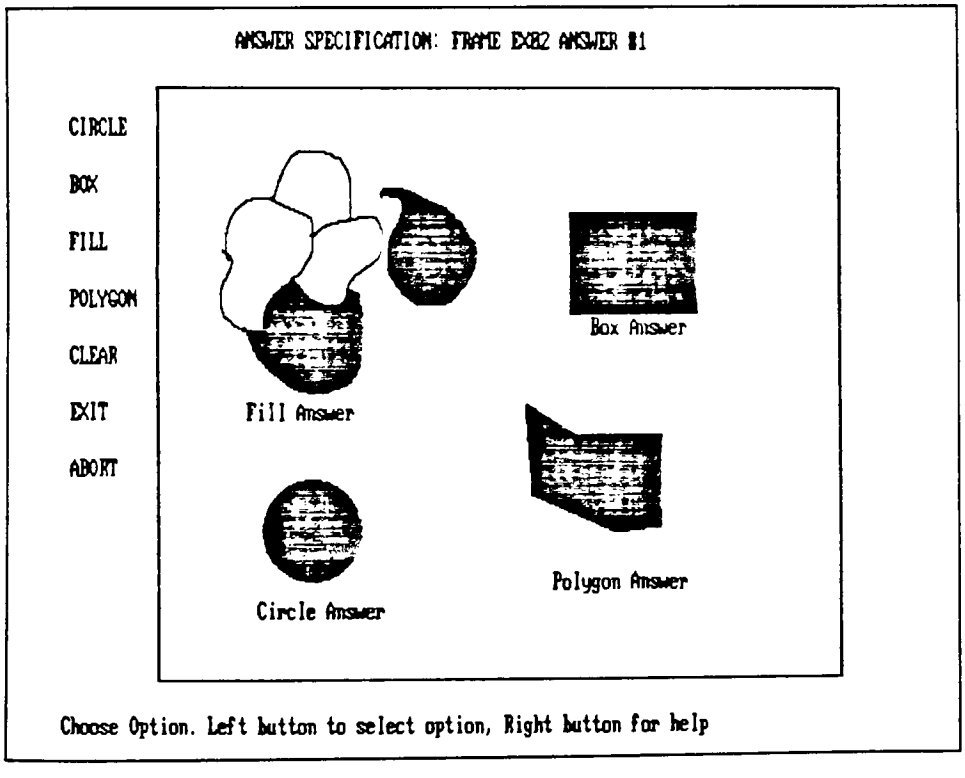
This option allows you to cancel the effects of this Frame Development session and leave Frame Development.

1. A prompt will be displayed asking if you wish to leave the Frame development without saving the modifications/additions done during the session.

2. Respond with Y if you wish to do so
   or with N if you wish to remain in Frame Development



Answer Specification screen before answer area specification



Answer Specification after specification of all answer area types

filename: ACIRCLE.HLP

Answer Specification
CIRCLE option

This option allows you to select a circle as an answer area.

1. Position the arrow at the center of the circle with the mouse
   or
   Press the right button of the mouse to abort the option

2. Press the left button of the mouse and a + sign marks the center of the chosen center.

3. Position the arrow on a point on the circumference of the circle.

4. Press the left button of the mouse and the circle is drawn on the frame and filled to identify it as an selected area.

filename: ABOX.HLP

Answer Specification
BOX option

This option allows you to specify a box (rectangle) as an answer area.

1. Position the arrow on the top left corner of the box with the mouse
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and a + sign marks the selected corner.

3. Move the arrow to the bottom right corner of the box and press the left button of the mouse.

4. A filled box is displayed to identify the selected area.

filename: AFILL.HLP

Answer Specification
FILL option

This option allows you to specify to area within a fully bounded shape as an answer area.

1. Move the arrow to a point within the shape
   or
   Press the right button of the mouse to abort the option.

2. Press the left button of the mouse and lthe shape is filled to identify the selected area.

Note: It is suggested when identifying answer areas that the fill area types be done first. The shape's boundaries should be boundaries that existed on the original frame. By doing the fill areas first this will avoid confusing answer area specifications as legitimate boundaries.

filename: APOLYGON.HLP

Answer Specification
POLYGON option

This option allows you to specify a polygon (up to ten sides) as an answer area.

1. Position the arrow on the first vertice of the polygon with the mouse.
   or
   Press the right button of the mouse to abort the option.

2. Press the left button to select the first vertice.

3. Move the arrow to the next vertice and press the left button to select it. The side of the polygon from the previous vertice to the current vertice is drawn.

4. Repeat step 3 until all vertices are specified.

---

5. Press the right button when complete. The polygon will be filled to identify it as a selected area.

filename: ACLEAR.HLP

Answer Specification
CLEAR option

This option allows you to erase the answer areas specified and remain in Answer Specification.

1. A prompt will be displayed to ask if your sure you want to clear the answer specifications.

2. Respond with Y if you do
   or with N if you don't.

filename: AEXIT.HLP

Answer Specification
EXIT option

This option allows you to save the answer specification.

1. A prompt will be displayed asking if you wish to save the answer specification and exit.

2. Respond with Y if you do
   or with N if you don't

filename: AABORT.HLP

Answer Specification
ABORT option

This option allows to leave Answer Specification without saving the current answer specification.

1. A prompt will be displayed asking you if you wish to leave Answer Specification without saving the current specification.

2. Respond with Y if you do
   or with N if you don't

APPENDIX B

This appendix describes the format of the stored data.

Note: Course_Step_Id is comprised of
- Course Id
- Step Type (Question, Lesson, or Option)
- Step Id

Step-Id is comprised of
- Step Type
- Step Id

STEPFILE.NDX format

Each key in the step index file has the following components:
- Course-Step-Id
- Sequence Error Flag

There will be only one occurrence of a key for each course step since each course step has only one Sequence Error Flag value at one time. A key for a question step or a lesson step has a data record in STEPFILE.DAT assigned to it.

STEPFILE.DAT format

A record in the step data file is in one of two formats:

For a question step it is comprised of:
- Step-Id of next step on success
- Step-Id of next step on failure
- Number of correct answers required for success
- Minimum number of questions asked
- Maximum number of questions asked
- Number of last questions asked used to identify success
- Maximum number of tries on a question

For a lesson step it is comprised of:
- Step-Id of next step

COMPFILE.NDX format

A key in the component index file can be in one of the following formats. The key-type identifies which information is in the key.

Start Step format: (*)
- key-type
- Course-Step-Id of the first step of the course

There is only one occurrence of this key for each course.

Question Group - Step format:
- key-type
- Question Group ID
- Course-Step-Id

There may be multiple keys for the same question group.

Question Group - Frame format:
key-type
Question Group ID
Frame ID

There may be multiple keys for the same question group.

Step Frame format: (*)
- key-type
- Course-Step-Id
- Frame ID

There is only one occurrence of this key for each course step.

Frame - Step format: (*)
key-type
Frame ID
- Course-Step-Id

There may be multiple keys for the same frame.

Frame Question Group format:
- key-type
- Frame ID
- Question Group ID

There may be multiple keys for the same frame.

Frame Existence format: (*)
- key-type
- Frame ID

Course Existence format: (*)
- key-type
- Course ID

Question Group Existence format:
- key-type
Question Group ID
- Report Flag

There is one occurrence of this key for each question group. Report Flag can only have one value for a specific question group at one time.

(*) indicates used in the current implementation.

ANOPFILE.NDX format

The key in the Answer/OptionChoice Index file can be one of the following two formats. The key-type identifies which format the key is in. Each key has a record in ANOPFILE.NDX assigned to it.

For answers:
- key-type
- Frame ID
- Answer Number

For Option Choices:
- key-type
- Course-Step-ID for option step
- Option Number

ANOPFILE.DAT format

Each record in the Answer/Option Data file is in one of two formats.

For answers:
- Answer Type (Right or Wrong)
- Answer Response

For option choices:
- Step-Id of next step

PERFFILE.NDX format

Each key in the student performance index file consists of
Student ID
Course-Step-Id

Answer Area Specification File Format (*.ANS)

Each record in the file consists of 6 integers.

The first record consists of:

- displacement to answer #1 specification
- displacement to answer #2 specification
- displacement to answer #3 specification
- displacement to answer #4 specification
- displacement to answer #5 specification
- displacement to end of file

A displacement value of 0 indicates that the answer number has no answer area specified for it.

The formats for the four answer area types are:

Circle: (1 record)

- answer number
  answer-type = CIRCLE
- lowest y coordinate of answer area
- highest y coordinate of answer area
- x coordinate of center of circle
- radius (horizontal pixels) of circle

Box: (1 record)

- answer number
- answer-type = BOX
- lowest y coordinate of answer area
  highest y coordinate of answer area
- lowest x coordinate of answer area
  highest x coordinate of answer area

Fill:

First record:

answer number

- answer-type = FILL
  lowest y coordinate for answer area
- highest y coordinate for answer area
  number of answer area specification points
- not used

Subsequent records contain coordinates of three points until all the answer area specification points are saved.

Polygon:

First record:

- answer number
- answer-type = POLYGON
- lowest y coordinate for the answer area
- highest y coordinate for the answer area
- number of answer area specification points
- not used

Subsequent records contain coordinates of three points until all the answer area specification points are saved.

APPENDIX C

This Appendix contains the original specifications for the Graphical Authoring System. It includes data store descriptions; process descriptions for creating, executing, and administrating a course; and the data dictionary.

DATA STORE DESCRIPTION

The following Entity-Relation Diagrams demonstrate the data and relationships required to support the Graphical Authoring System. It is not intended to be a database design. (See Appendix for brief description of the Entity-Relation Diagram notation.) Note that some entities may appear in more than one diagram.

The following table summarizes how the 3 major components use the data stores.

ENTITY-RELATION DIAGRAM	CREATE COURSE	ADMINISTRATE COURSE	EXECUTE COURSE
COURSE-COMPONENTS QUESTION-STEP-CONTENTS LESSON-STEP-CONTENTS OPTION-STEP-CONTENTS	UPDATE		READ
STUDENT-PERFORMANCE		READ/DELETE	UPDATE
CURRICULUM		UPDATE	READ

Additional information accompanies each of the diagrams to further describe the relations and the constraints upon them. See the data dictionary for further descriptions of the contents of the entities. When describing access/update of stored data, the keys to the entities may be used to describe relationships between entities or to identify the entity in order to indicate changes in non-key attributes.

Table of entity keys:

Entity Name	Key
COURSE	COURSE-ID
COURSE-STEP	COURSE-STEP-ID + COURSE-REF
QUESTION-STEP	QUESTION-STEP-ID + COURSE-REF
LESSON-STEP	LESSON-STEP-ID + COURSE-REF
OPTION-STEP	OPTION-STEP-ID + COURSE-REF
LESSON-FRAME	LESSON-FRAME-ID
OPTION-FRAME	OPTION-FRAME-ID
OPTION-CHOICE	OPTION-NBR + OPTION-STEP-REF + COURSE-STEP-REF
QUESTION-GROUP	QUESTION-GROUP-ID
QUESTION	QUESTION-ID
ANSWER	ANSWER-NBR + QUESTION-REF
STUDENT	STUDENT-ID
QUESTION-INSTANCE	STUDENT-REF + QUESTION-REF
COURSE-COMPLETION	STUDENT-REF + COURSE-REF
CURRENT-STEP	STUDENT-REF + COURSE-STEP-REF

COURSE COMPONENTS

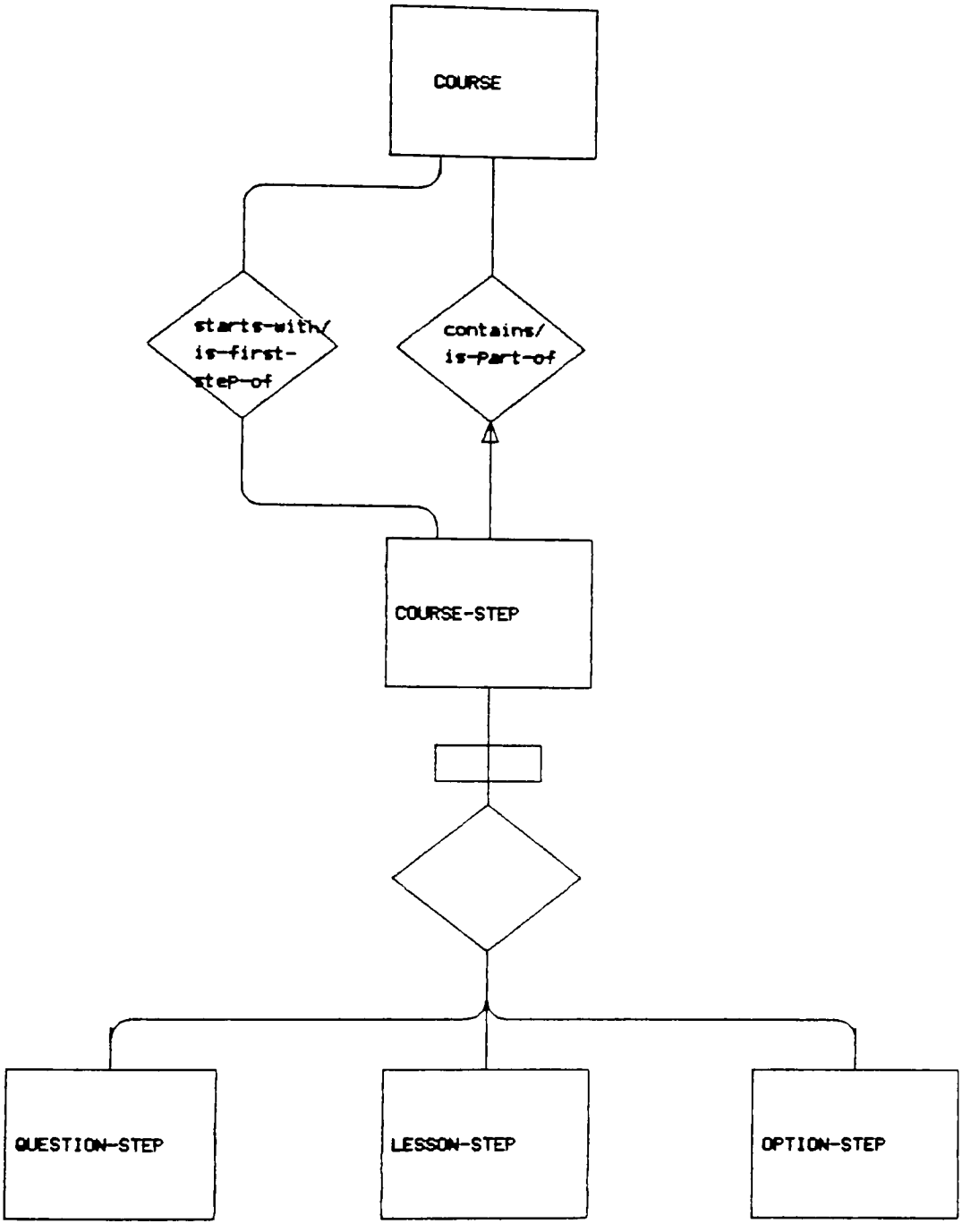
For each COURSE there is at most one COURSE-STEP that is specified as the starting COURSE-STEP. After the development of a valid course is completed, there is one COURSE-STEP that is specified as the starting step of the COURSE.

COURSE starts-with COURSE-STEP (1:1)
COURSE-STEP is-first-step-of COURSE (1:1)
or (example of using keys)
COURSE-ID starts-with COURSE-STEP-ID
COURSE-STEP-ID is-first-step-of COURSE-ID

Each COURSE is made up of COURSE-STEPs. After a valid course is developed there is at least one COURSE-STEP in the COURSE.

COURSE consists-of COURSE-STEP (1:N)
COURSE-STEP is-part-of COURSE (1:1)

A COURSE-STEP is one of three types -- QUESTION-STEP, LESSON-STEP or OPTION-STEP



COURSE-COMPONENTS
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 04-SEP-86

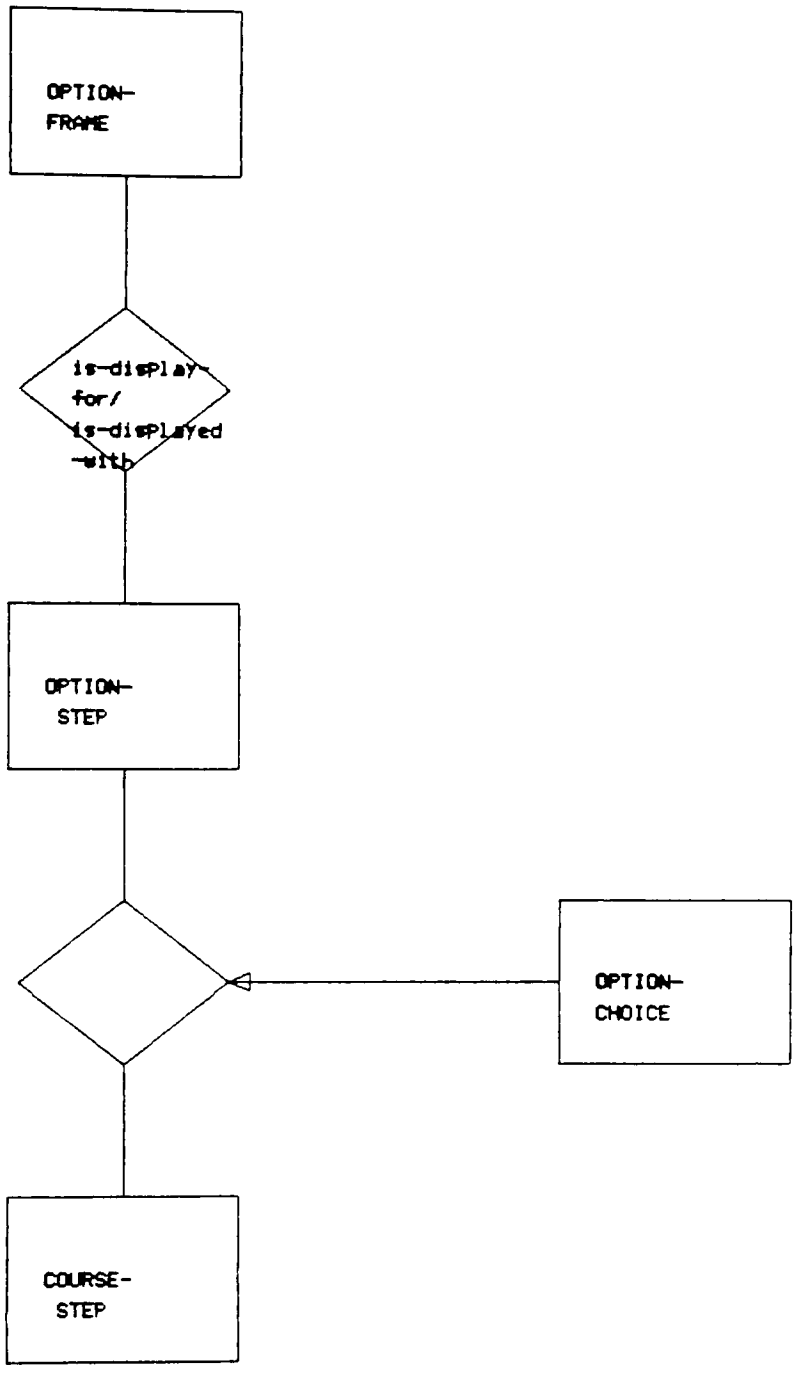
OPTION STEP CONTENT

Each OPTION-STEP has a screen display representation (OPTION-FRAME). An OPTION-FRAME can be used by multiple OPTION-STEPs.

OPTION-STEP is-displayed-with OPTION-FRAME (1:1)
OPTION-FRAME is-display-for OPTION-STEP (1:N)

For an OPTION-STEP multiple areas can be identified as OPTION-CHOICEs that specify what the next COURSE-STEP is. When a valid course has been developed, each OPTION-CHOICE requires that an associated OPTION-STEP and next COURSE-STEP have been defined.

OPTION-CHOICE for OPTION-STEP specifies the next COURSE-STEP



OPTION-STEP-CONTENT
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 05-SEP-86

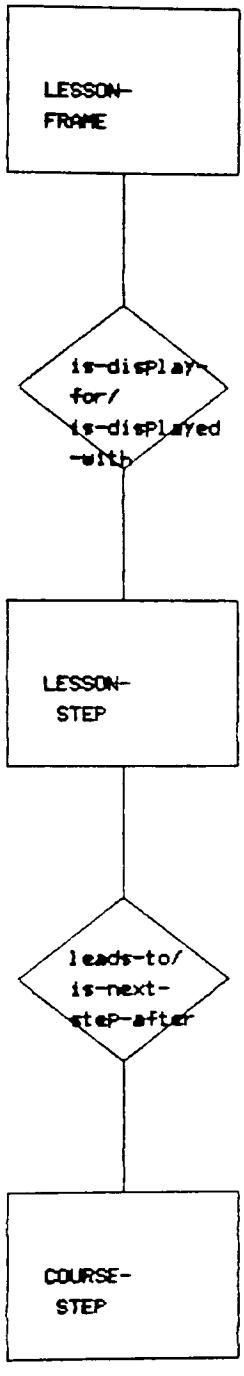
LESSON STEP CONTENT

Each LESSON-STEP has a screen display representation (LESSON-FRAME). A LESSON-FRAME can be used by multiple LESSON-STEPs.

LESSON-STEP is-displayed-with LESSON-FRAME (1:1)
LESSON-FRAME is-display-for LESSON-STEP (1:N)

Each LESSON-STEP leads to another COURSE-STEP.

LESSON-STEP leads-to COURSE-STEP(1:1)
COURSE-STEP is-next-step-after LESSON-STEP (1:N)



LESSON-STEP-CONTENT
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 05-SEP-86

QUESTION STEP CONTENT

Each QUESTION-STEP has one group of questions to utilize. A QUESTION-GROUP is-used-by one or more QUESTION-STEPs.

QUESTION-STEP uses QUESTION-GROUP (1:1)
QUESTION-GROUP is-used-by QUESTION-STEP (1:N)

Each QUESTION can be a member of one or more QUESTION-GROUPs but the QUESTION does not necessarily have to be identified as contained in any QUESTION-GROUP. A QUESTION-GROUP contains one or more QUESTIONs.

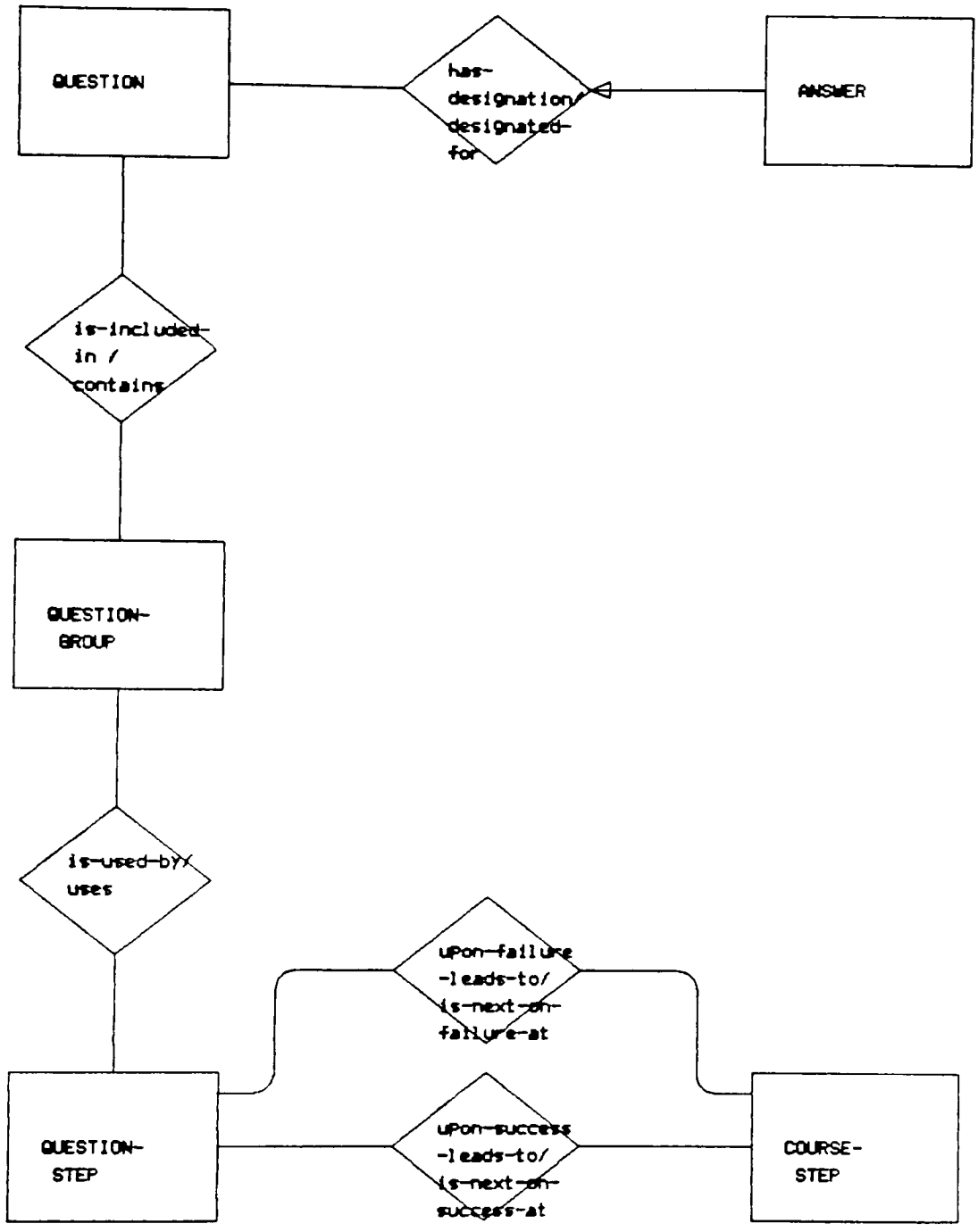
QUESTION-GROUP contains QUESTION (1:N)
QUESTION is-included-in QUESTION-GROUP (1:N)

Each QUESTION is assigned one or more ANSWER specifications. An ANSWER specification is assigned to a particular QUESTION.

QUESTION is-assigned ANSWER (1:N)
ANSWER is-specified-for QUESTION (1:1)

For each QUESTION-STEP the next COURSE-STEP is specified one for successful completion of the QUESTION-STEP and one for failure at QUESTION-STEP.

QUESTION-STEP on-success-leads-to COURSE-STEP (1:1)
COURSE-STEP is-next-step-on-success-at QUESTION-STEP (1:N)
QUESTION-STEP on-failure-leads-to COURSE-STEP (1:1)
COURSE-STEP is-next-step-on-failure-at COURSE-STEP (1:N)



QUESTION-STEP-CONTENT
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 08-SEP-86

STUDENT PERFORMANCE

Each time a STUDENT is asked a QUESTION this occurrence is saved.

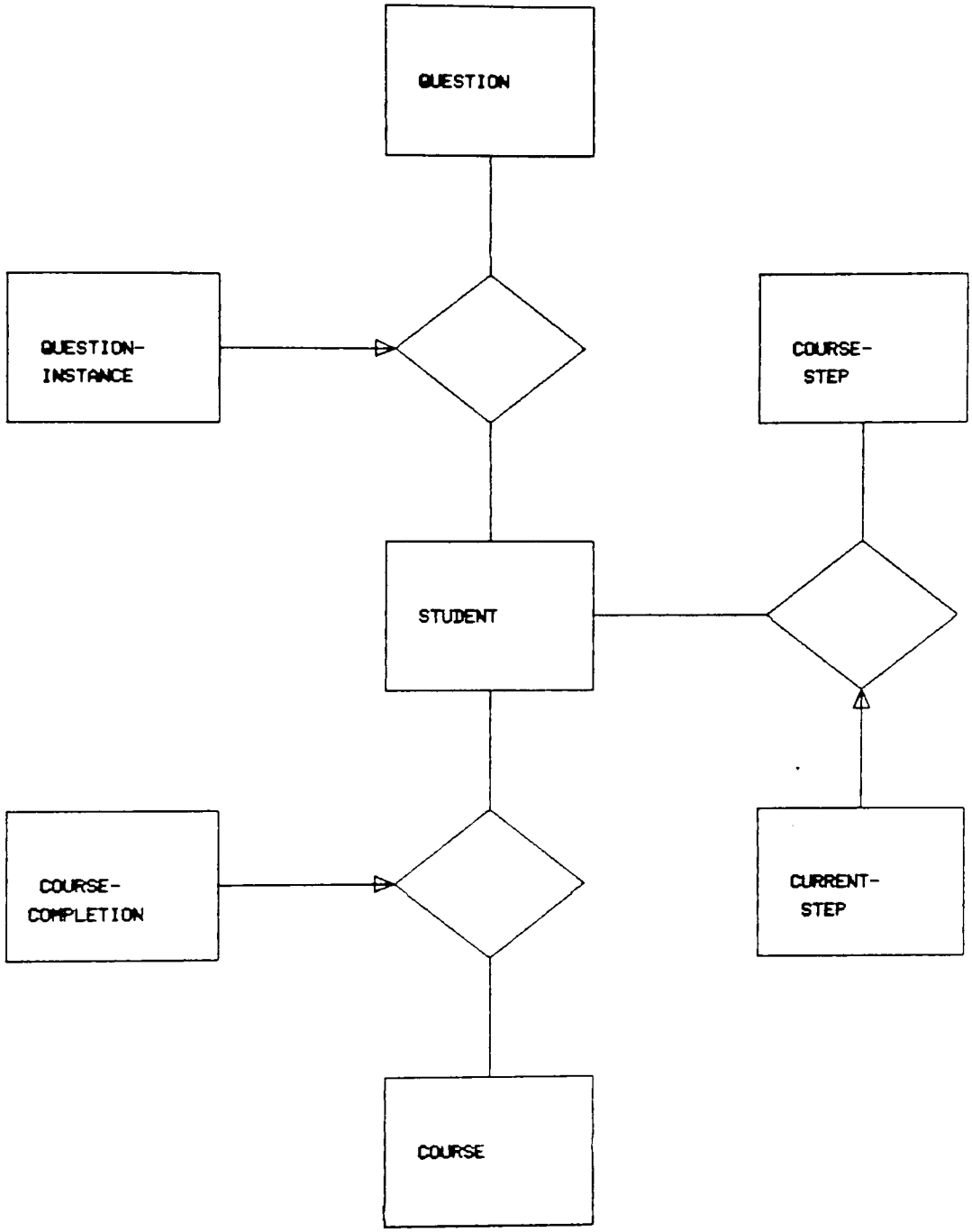
QUESTION-INSTANCE for STUDENT having-been-presented QUESTION

Each time a STUDENT completes a COURSE information about this occurrence is saved.

COURSE-COMPLETION of COURSE by STUDENT

Each time a STUDENT begins a COURSE-STEP, his CURRENT-STEP in the COURSE is updated.

CURRENT-STEP in COURSE for STUDENT is COURSE-STEP

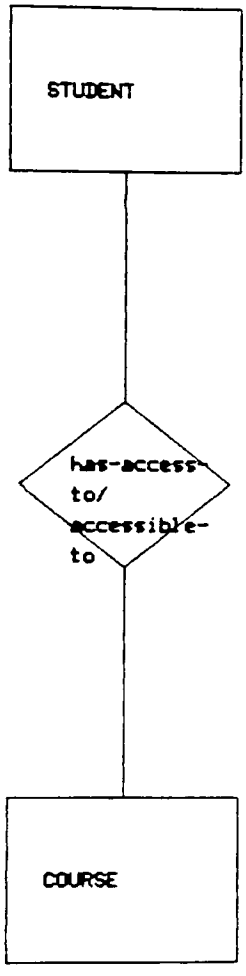


STUDENT-PERFORMANCE
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 08-SEP-86

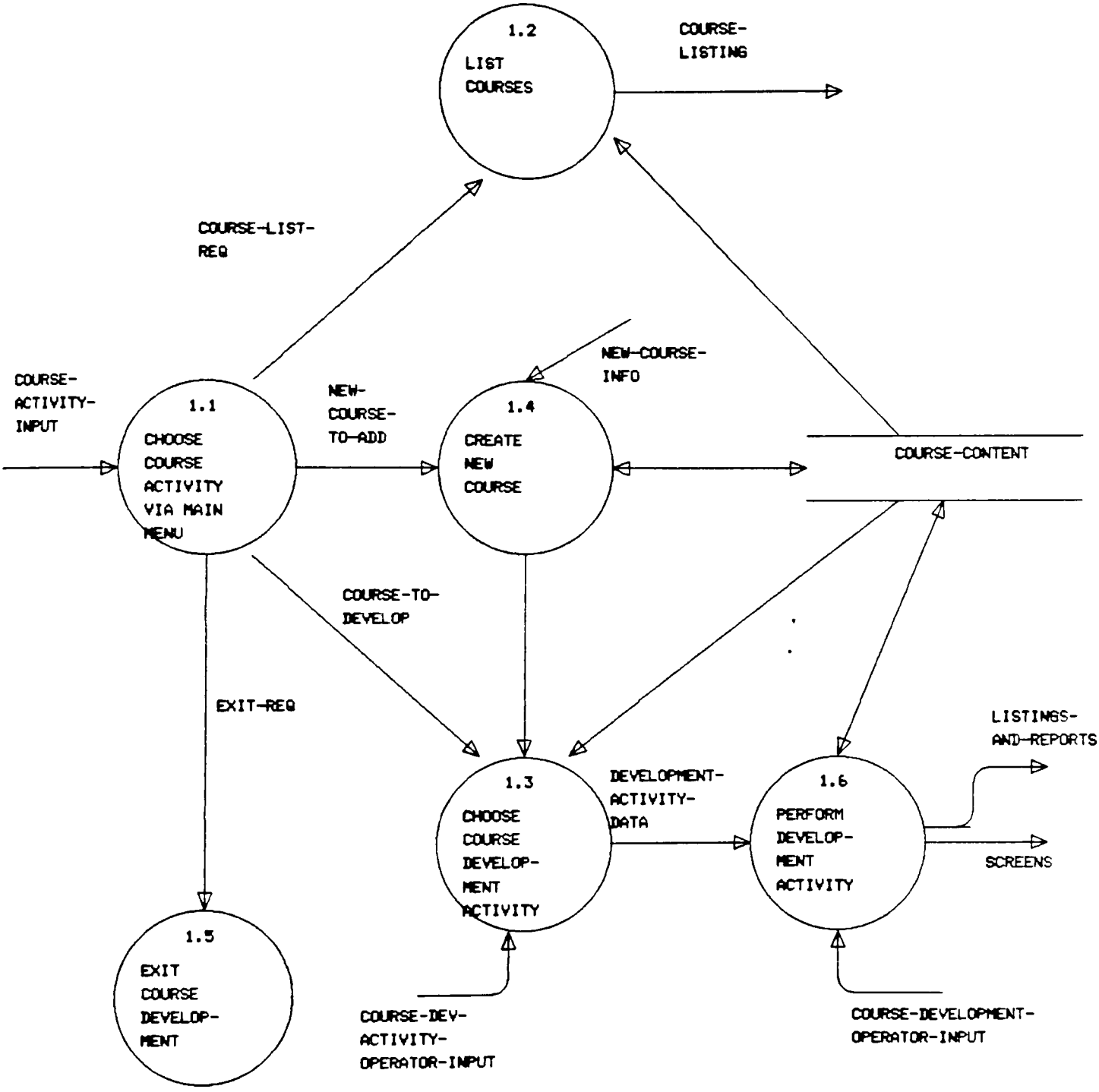
CURRICULUM

At a given time, a STUDENT is assigned to have access to a set of COURSEs

STUDENT has-access-to COURSE (1:N)
COURSE is-accessible-to STUDENT (1:N)



CURRICULUM
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 05-SEP-86



1.0 CREATE COURSE
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 10-OCT-86

1.0 CREATE COURSE.

1.1 CHOOSE COURSE ACTIVITY VIA MAIN MENU.
For each MENU-XFER to the main menu:
Display main menu that includes activities listed below,
Obtain MENU-CHOICE from the developer,
If the MENU-CHOICE is invalid
Display appropriate SCREEN-MSG and go back to get input again,
For the activity specified by the MENU-CHOICE:
Display prompts for the inputs specified in the table,
Obtain inputs from the developer,
If the exit key was pressed
Go back and get MENU-CHOICE
If the input is invalid
Display an appropriate SCREEN-MSG and get input again,
Create COURSE-ACTIVITY-DATA

Table 1-1

ACTIVITY	INPUTS	VALIDATION	COURSE-ACTIVITY-DATA
LIST COURSES	none		COURSE-LIST-REQ
DEVELOP EXISTING COURSE	COURSE-ID	COURSE EXISTS	COURSE-TO-DEVELOP
DEVELOP NEW COURSE	COURSE-ID	COURSE DOESN'T EXIST	NEW-COURSE-TO-DEVELOP
EXIT	none		EXIT-REQ

1.2 LIST COURSES.
For each COURSE-LIST-REQ:
Print COURSE-LIST-HEADER,
For all COURSEs that exist
Print COURSE-LIST-LINE sorted in COURSE-ID order
Display SCREEN-MSG of "Course List Complete. Press key to continue."
Obtain keystroke from developer,
Create MENU-XFER
END OF PROCESS

1.3 CHOOSE COURSE DEVELOPMENT ACTIVITY.
For each COURSE-TO-DEVELOP or MENU-XFER:
Support the activity hierarchy identified in the following table,
For the appropriate activity :
Display menu or prompt(s) for the input field(s),
Obtain MENU-CHOICE or input field values from the developer,
If the exit key is pressed
Create MENU-XFER to next higher menu
If an input is invalid
Display an appropriate SCREEN-MSG and go back to get input,
Create DEVELOPMENT-ACTIVITY-DATA

Table 1-3

ACTIVITY	INPUTS	DEVELOPMENT-ACTIVITY-DATA	I.S.
DEVELOP COURSE	MENU-CHOICE(1)	none	
**QUESTION STEP	MENU-CHOICE (1)	none	
****ADD	QUESTION-STEP-ID(2)	QUESTION-STEP-TO-ADD	a,b
****MODIFY	QUESTION-STEP-ID(3)	QUESTION-STEP-TO-MODIFY	b
****DISPLAY	QUESTION-STEP-ID(3)	QUESTION-STEP-TO-DISPLAY	a,b
****LIST	None	COURSE-QUESTION-STEPS-TO-LIST	e
****DELETE	QUESTION-STEP-ID(3)	QUESTION-STEP-TO-DELETE	a,b
****DELETE ALL	QUESTION-STEP-ID(3)	ENTIRE-QUESTION-STEP-TO-DELETE	e
**QUESTION GROUP	MENU-CHOICE(1)	none	
****ADD	QUESTION-GROUP-ID(4)	QUESTION-GROUP-TO-ADD	b
****ASSIGN QUESTION	QUESTION-GROUP-ID(5) QUESTION-ID(5)	QUESTION-TO-ASSIGN-GROUP	b
****ASSIGN TO STEP	QUESTION-GROUP-ID(5) QUESTION-STEP-ID(3)	QUESTION-GROUP-TO-ASSIGN-TO-STEP	b
****LIST	none	QUESTION-GROUP-LISTING-REQ	e
****DELETE	QUESTION-GROUP-ID(5)	QUESTION-GROUP-TO-DELETE	b
****DELETE ALL	QUESTION-GROUP-ID(5)	ENTIRE-QUESTION-GROUP-TO-DELETE	d
****REMOVE QUEST	QUESTION-GROUP-ID(5) QUESTION-ID(5)	QUESTION-TO-REMOVE-FROM-GROUP	b
**QUESTION	MENU-CHOICE(1)	none	
****ADD	QUESTION-ID(4)	QUESTION-TO-ADD	a
****ADD TO GROUP	QUESTION-ID(5) QUESTION-GROUP-ID(5)	QUESTION-TO-ADD-TO-GROUP	b
****MODIFY	QUESTION-ID(5)	QUESTION-TO-MODIFY	a
****DISPLAY	QUESTION-ID(5)	QUESTION-TO-DISPLAY	a
****LIST IN GROUP	QUESTION-GROUP-ID(5)	QUESTION-GROUP-TO-LIST	e
****DELETE	QUESTION-ID(5)	QUESTION-TO-DELETE	a
**ANSWER	MENU-CHOICE(1)	none	
****ADD	QUESTION-ID(5) RIGHT-WRONG-SW	ANSWER-AREA-TO-ADD	a
****MODIFY RESPONSE	QUESTION-ID(5)	ANSWER-AREA-TO-MODIFY	a
****DISPLAY	QUESTION-ID(5)	ANSWER-AREA-TO-DISPLAY	a
****LIST	QUESTION-ID(5)	ANSWER-AREA-TO-LIST	e
****DELETE	QUESTION-ID(5)	ANSWER-AREA-TO-DELETE	a
**LESSON STEP	MENU-CHOICE(1)	none	
****ADD	LESSON-STEP-ID(2)	LESSON-STEP-TO-ADD	d
****MODIFY	LESSON-STEP-ID(3)	LESSON-STEP-TO-MODIFY	d
****DISPLAY	LESSON-STEP-ID(3)	LESSON-STEP-TO-DISPLAY	d
****LIST	none	COURSE-LESSON-STEPS-TO-LIST	e
****DELETE	LESSON-STEP-ID(3)	LESSON-STEP-TO-DELETE	d
****DELETE ALL	LESSON-STEP-ID(3)	ENTIRE-LESSON-STEP-TO-DELETE	e
**LESSON-FRAME	MENU-CHOICE(1)	none	
****ADD	LESSON-FRAME-ID(4)	LESSON-FRAME-TO-ADD	d
****MODIFY	LESSON-FRAME-ID(5)	LESSON-FRAME-TO-MODIFY	d
****ASSIGN TO STEP	LESSON-FRAME-ID(5) LESSON-STEP-ID(5)	LESSON-FRAME-ASSIGNMENT	d
****DISPLAY	LESSON-FRAME-ID(5)	LESSON-FRAME-TO-DISPLAY	d
****LIST	none	LESSON-FRAME-LIST-REQ	e
****DELETE	LESSON-FRAME-ID(5)	LESSON-FRAME-TO-DELETE	d

ACTIVITY	INPUTS	DEVELOPMENT-ACTIVITY-DATA	I.S.
**OPTION STEP	MENU-CHOICE(1)		
****ADD	OPTION-STEP-ID(2)	OPTION-STEP-TO-ADD	c
****MODIFY	OPTION-STEP-ID(3)	OPTION-STEP-TO-MODIFY	c
****DISPLAY	OPTION-STEP-ID(3)	OPTION-STEP-TO-DISPLAY	c
****LIST	none	COURSE-OPTION-STEP-LIST-REQ	e
****DELETE	OPTION-STEP-ID(3)	OPTION-STEP-TO-DELETE	c
****DELETE ALL	OPTION-STEP-ID(3)	ENTIRE-OPTION-STEP-TO-DELETE	e
**OPTION FRAME	MENU-CHOICE(1)		
****ADD	OPTION-FRAME-ID(4)	OPTION-FRAME-TO-ADD	c
****MODIFY	OPTION-FRAME-ID(5)	OPTION-FRAME-TO-MODIFY	c
****ASSIGN TO STEP	OPTION-FRAME-ID(5) OPTION-STEP-ID(3)	OPTION-FRAME-TO-ASSIGN-TO-STEP	c
****DISPLAY	OPTION-FRAME-ID(5)	OPTION-FRAME-TO-DISPLAY	c
****LIST	none	OPTION-FRAME-LIST-REQ	e
****DELETE	OPTION-FRAME-ID(5)	OPTION-FRAME-TO-DELETE	c
**OPTION AREA	MENU-CHOICE(1)		
****ADD	OPTION-STEP-ID(3) COURSE-STEP-ID	OPTION-AREA-TO-ADD	c
****MODIFY	OPTION-STEP-ID(3) COURSE-STEP-ID(3)	OPTION-AREA-TO-MODIFY	c
****DISPLAY	OPTION-STEP-ID(3) COURSE-STEP-ID(3)	OPTION-AREA-TO-DISPLAY	c
****LIST	OPTION-STEP-ID	OPTION-AREA-LIST-REQ	e
****DELETE	OPTION-STEP-ID(3) COURSE-STEP-ID(3)	OPTION-AREA-TO-DELETE	c
****DELETE ALL	OPTION-STEP-ID(3)	ALL-OPTION-AREAS-TO-DELETE	e
**STARTING STEP	MENU-CHOICE	none	
****ADD/MODIFY	COURSE-STEP-ID	STARTING-STEP-FOR-COURSE	a
****DISPLAY	none	STARTING-STEP-DISPLAY-REQ	a
**REPORTS	MENU-CHOICE(1)		
****CONTENT	none	CONTENT-RPT-REQ	*
****SEQUENCE	none	SEQUENCE-RPT-REQ	*

Input Validations:

(1) Valid menu choice
(2) Course step does not currently exist
(3) Course step currently exists
(4) ID does not currently exist
(5) ID currently exists

I.S. column indicates the implementation strategy:

(a) Step 1: Implement question step with one question (thatis, question group is not implemented)
(b) Step 2: Implement question group.
(c) Step 3: Implement option step.
(d) Step 4: Implement lesson step.
(e) Step 5: Implement List and Delete All options and add "administrate course" capabilities.
(*) During steps 1 - 5, implement the reporting that supports what has been implemented.

For each step, implement the corresponding portion of "execute course".

Minimally, complete step 1.

1.4 CREATE NEW COURSE

For each NEW-COURSE-TO-DEVELOP:

Display prompt for "Enter Course Title"
Obtain COURSE-TITLE from developer,
Add COURSE for COURSE-ID,
Create COURSE-TO-DEVELOP

END OF PROCESS.

1.5 FINISH COURSE DEVELOPMENT

For each EXIT-REQ:

Do any neccessary cleanup

END OF PROCESS.

1.6 PERFORM COURSE DEVELOPMENT ACTIVITY.

For each COURSE-DEVELOPMENT-ACTIVITY-DATA:

The activity is performed as specified in the following process descriptions.
When the process is complete or the exit key is pressed, a MENU-XFER to the invoking menu is created.

For each QUESTION-STEP-TO-ADD:

Display QUESTION-STEP-PARAMETER-SCREEN with the
QUESTION-STEP-PARAMETERS, SUCCESS-COURSE-STEP and
FAILURE-COURSE-STEP fields blank,
Get QUESTION-STEP-PARAMETERS, SUCCESS-COURSE-STEP, and
FAILURE-COURSE-STEP from the developer,
Display an appropriate SCREEN-MSG and get the values from developer again for the following:

NBR-CORRECT-ANS-REQUIRED > MIN-QUEST-ASKED,
MIN-QUEST-ASKED > MAX-QUEST-ASKED,
NBR-OF-LAST-QUEST > MAX-QUEST-ASKED,
NBR-CORRECT-ANS-REQUIRED > NBR-LAST-QUEST

Add QUESTION-STEP for QUESTION-STEP-ID and specify the QUESTION-STEP is part-of COURSE-ID,
If a QUESTION-GROUP-ID was entered

If the entered QUESTION-GROUP-ID does not exist

Create QUESTION-GROUP,

Specify that QUESTION-GROUP uses QUESTION-STEP,

If a SUCCESS-COURSE-STEP was entered

If the entered SUCCESS-COURSE-STEP does not exist

Create COURSE-STEP for SUCCESS-COURSE-STEP with attributes "unspecified",

Specify that QUESTION-STEP upon-success-leads-to SUCCESS-COURSE-STEP,

If a FAILURE-COURSE-STEP was entered

If the entered FAILURE-COURSE-STEP does not exist

Create COURSE-STEP for FAILURE-COURSE-STEP with attributes "unspecifies",

Specify QUESTION-STEP upon-failure-leads-to FAILURE-COURSE-STEP,
Display a SCREEN-MSG of "Question Step Add Complete. Press key to continue"
Obtain keystroke from developer.
END OF PROCESS.

For each QUESTION-STEP-TO-MODIFY:
Display QUESTION-STEP-PARAMETER-SCREEN with values from QUESTION-STEP for QUESTION-STEP-ID,
Obtain QUESTION-STEP-PARAMETERS, SUCCESS-COURSE-STEP and FAILURE-COURSE-STEP from developer,
Display appropriate SCREEN-MSG and get values from the developer again for the following:
<see validations for QUESTION-STEP-TO-ADD>
Update QUESTION-STEP for QUESTION-STEP-ID,
Update specification of next-step-on-success, next-step-on-failure and QUESTION-GROUP used if they have changed,
Display SCREEN-MSG of "Question Step Modify Complete. Press key to continue"
Obtain keystroke from developer.
END OF PROCESS

For each QUESTION-STEP-TO-DISPLAY:
Display QUESTION-STEP-PARAMETER-SCREEN with the current values from QUESTION-STEP for QUESTION-STEP-ID,
Display SCREEN-MSG of "Question Step Display Complete. Press key to continue"
Obtain keystroke from developer.
END OF PROCESS

For each COURSE-QUESTION-STEPS-TO-LIST:
Print QUESTION-STEP-LISTING-HEADER with COURSE-ID
For all QUESTION-STEPs that are part-of COURSE-ID :
Sort by QUESTION-STEP-ID,
Determine QUESTION-GROUP-ID used-by QUESTION-STEP-ID,
Print QUESTION-STEP-LISTING-LINE
Display SCREEN-MSG of "Question Step Listing Complete. Press key to continue",
Obtain keystroke from developer
END OF PROCESS

For each QUESTION-STEP-TO-DELETE:
Display SCREEN-MSG of "Are you sure you want to delete?"
Obtain DELETE-RESPONSE from developer,
If DELETE-RESPONSE = "Y"
Delete the QUESTION-STEP for QUESTION-STEP-ID (QUESTION-GROUPs associated to QUESTION-STEP remain),
Display SCREEN-MSG of "Question Step Delete Complete. Press key to continue",
Obtain keystroke from developer
END OF PROCESS

For each ENTIRE-QUESTION-STEP-TO-DELETE:
Display SCREEN-MSG of "Are you sure you want to delete?"
Obtain DELETE-RESPONSE from developer,
If DELETE-RESPONSE = "Y"
For each QUESTION-GROUP which is used-by QUESTION-STEP-ID:

```
                    If the QUESTION-GROUP is not used-by another QUESTION-STEP
                        For each QUESTION included-in QUESTION-GROUP:
                            If QUESTION is not included-in another QUESTION-GROUP
                                Delete the QUESTION and associated ANSWERs
                        Delete QUESTION-GROUP
                Delete QUESTION-STEP,
                Display SCREEN-MSG of "Entire Question Step Delete. Press key to continue",
                Obtain keystroke from developer
END OF PROCESS


For each QUESTION-GROUP-TO-ADD:
        Add QUESTION-GROUP for QUESTION-GROUP-ID,
        Display SCREEN-MSG of "Question Group Add Complete. Press key to continue",
        Obtain keystroke from developer
END OF PROCESS

For each QUESTION-TO-ASSIGN-GROUP:
        Specify QUESTION-ID is included-in QUESTION-GROUP-ID,
        Display SCREEN-MSG of "Assign Question to Group Complete. Press key to continue",
        Obtain keystroke from developer
END OF PROCESS


For each QUESTION-GROUP-TO-ASSIGN-TO-STEP:
        Specify QUESTION-GROUP-ID is used-by QUESTION-STEP-ID,
        Display SCREEN-MSG of "Assign Question Step to Group Complete. Press key to
        continue",
        Obtain keystroke from developer
END OF PROCESS


For each QUESTION-GROUP-LISTING-REQ:
        Print QUESTION-GROUP-LISTING-HEADER,
        For all QUESTION-GROUPs that are used-by QUESTION-STEP which are part of
        COURSE-ID:
                Sort by QUESTION-GROUP-ID,
                Print QUESTION-GROUP-LISTING-LINE for each QUESTION-GROUP-ID
        Display SCREEN-MSG of "Question Group Listing Done. Press key to continue",
        Obtain keystroke from developer
END OF PROCESS


For each QUESTION-GROUP-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete?"
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                If QUESTION-GROUP uses only QUESTION-STEPs that are contained-in
                COURSE-ID or QUESTION-GROUP is not used-by any QUESTION-STEP
                        Delete QUESTION-GROUP for QUESTION-GROUP-ID,
                Else
                        Remove that QUESTION-GROUP is used-by any QUESTION-STEP
                        contained-in COURSE-ID,
                Display SCREEN-MSG of "Question Group Deleted. Press key to continue",
                Obtain keystroke from developer
END OF PROCESS
```

```
For each ENTIRE-QUESTION-GROUP-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete?"
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                For each QUESTION which is included-in QUESTION-GROUP-ID:
                        If QUESTION is not included-in another QUESTION-GROUP
                                Delete QUESTION
                If QUESTION-GROUP uses only QUESTION-STEPs that are contained-in
                COURSE-ID or QUESTION-GROUP is not used-by any QUESTION-STEP
                        Delete QUESTION-GROUP
                Else
                        Remove that QUESTION is used-by any QUESTION-STEP contained-in
                        COURSE-ID
                Display SCREEN-MSG of "Question Group Deleted. Press key to continue".
                Obtain keystroke from developer
END OF PROCESS


For each QUESTION-TO-REMOVE-FROM-GROUP:
        Delete specification of QUESTION included-in QUESTION-GROUP
        Display SCREEN-MSG of "Removal from Question Group Complete. Press key to
        continue"
END OF PROCESS


For each QUESTION-TO-ADD:
        Display QUESTION-DEVELOP-SCREEN for QUESTION-ID with blank graphics area,
        Accept and process the developer's graphic directives as described in the General
        Specifications for Graphic Capabilities to develop new question,
END OF PROCESS


For each QUESTION-TO-MODIFY:
        Display QUESTION-DEVELOP-SCREEN for QUESTION-ID with the the GRAPHICS for
        the QUESTION-ID,
        Accept and process the developer's graphic directives as described in the General
        Specifications for Graphic Capabilities to modify a question
END OF PROCESS


For each QUESTION-TO-ADD-TO-GROUP:
        Display QUESTION-DEVELOP-SCREEN for QUESTION-ID with blank graphics area,
        Accept and process the developer's graphic directives as described in the General
        Specifications for Graphic Capabilities to develop a new question and assign to group.
END OF PROCESS


For each QUESTION-TO-DISPLAY:
        Display DISPLAY-QUESTION-SCREEN with GRAPHICS for QUESTION-ID,
        Display SCREEN-MSG of "Question Display Complete. Press key to continue",
        Obtain keystroke from developer
END OF PROCESS


For each QUESTION-GROUP-TO-LIST:
        Print QUESTION-LISTING-HEADER,
```

```
        For all QUESTIONs included-in QUESTION-GROUP-ID:
                Sort them by QUESTION-ID,
                Print QUESTION-LIST-LINE for each QUESTION-ID
        Display SCREEN-MSG of "Listing Complete. Press key to continue",
        Obtain keystroke from developer
END OF PROCESS


For each QUESTION-TO-DELETE:
        If QUESTION-ID is contained-in other than COURSE-ID (via QUESTION-GROUP and
        QUESTION-STEP)
                Display SCREEN-MSG of "Question used in another course. Do you want to
                delete?"
        Else
                Display SCREEN-MSG of "Are you sure you want to delete?"
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                Delete all ANSWERs that are designated-to QUESTION-ID,
                Delete QUESTION for QUESTION-ID,
                Display SCREEN-MSG of "Question Delete Complete. Press key to continue",
                Obtain keystroke from developer
END OF PROCESS


For each ANSWER-AREA-TO-ADD:
        If GRAPHICS for QUESTION-ID is "unspecified"
                Display SCREEN-MSG of "Question not yet specified. Answer area can not be
                added. Press key to continue.",
                Obtain keystroke from developer,
                END OF PROCESS
        Display SCREEN-MSG of "Please specify answer response",
        Obtain ANSWER-RESPONSE from developer,
        If RIGHT-WRONG-SW = "Wrong"
                Display SCREEN-MSG of "Do you wish to specify answer area [A] or default
                response [D]?",
                Obtain AREA-TYPE from developer,
                If AREA-TYPE is "D"
                        Add ANSWER with ANSWER-AREA of "Default" and assign it to
                        QUESTION-ID
        If answer area is to be specified (RIGHT-WRONG-SW = "Right" or (RIGHT-WRONG-SW
        = "Wrong" and AREA-TYPE = "A"))
                Display ANSWER-AREA-SCREEN with GRAPHICS for QUESTION-ID and no
                answer area displayed,
                Accept and process developer's graphic directives as specified in General
                Specification for Graphic Capabilities to develop answer area,
                If an answer area was not defined (exit with no save from graphic development)
                        END OF PROCESS
        Add ANSWER assigned to QUESTION-ID with unique ANSWER-NBR, and
        ANSWER-RESPONSE and ANSWER-AREA as specified in this process, and
        ANSWER-TYPE = RIGHT-WRONG-SW,
        Display SCREEN-MSG of "Answer Add Complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each ANSWER-AREA-TO-MODIFY:
        For each ANSWER assigned to QUESTION-ID:
                Display ANSWER-LIST-LINE
```

```
        Display SCREEN-MSG of "Select Answer Number"
        Obtain ANSWER-NBR from developer,
        Display prompt for new ANSWER-RESPONSE for ANSWER-NBR,
        Obtain ANSWER-RESPONSE from developer,
        Update ANSWER for ANSWER-NBR with ANSWER-RESPONSE,
        Display SCREEN-MSG of "Answer response modify complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each ANSWER-AREA-TO-DISPLAY:
        For each ANSWER assigned to QUESTION-ID:
                Display ANSWER-LIST-LINE
        Display SCREEN-MSG of "Select Answer Number",
        Obtain ANSWER-NBR from developer,
        Display ANSWER-AREA-SCREEN with GRAPHICS for QUESTION-ID and the
        ANSWER-AREA for ANSWER-NBR shaded,
        Display SCREEN-MSG of "Answer Area Display Complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each ANSWER-AREA-TO-LIST:
        Print ANSWER-LIST-HEADER,
        For each ANSWER assigned to QUESTION-ID:
                Print ANSWER-LIST-LINE
        Display SCREEN-MSG of "Answer List Complete. Press key to continue."
        Obtain keystroke from the developer
END OF PROCESS


For each ANSWER-AREA-TO-DELETE:
        For each ANSWER assigned to QUESTION-ID:
                Display ANSWER-LIST-LINE
        Display SCREEN-MSG of "Select Answer Number",
        Obtain ANSWER-NBR from developer,
        Display SCREEN-MSG of "Are you sure you want to delete?",
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                Delete ANSWER for ANSWER-NBR which is assigned to QUESTION-ID
                Display SCREEN-MSG of "Answer Delete Complete. Press key to continue.",
                Obtain keystroke from developer
END OF PROCESS


For each LESSON-STEP-TO-ADD:
        Display LESSON-STEP-PARAMETER-SCREEN with blank fields,
        Obtain NEXT-COURSE-STEP (optional) and
                LESSON-FRAME-ID (optional) from the developer,
        Validate the values of the inputs and go back to get input again if there is an invalid
        value,
        Add LESSON-STEP for LESSON-STEP-ID and specify it is part-of COURSE-ID,
        If NEXT-COURSE-STEP is entered
                If NEXT-COURSE-STEP is not currently part-of COURSE-ID
                        Add COURSE-STEP for NEXT-COURSE-STEP with attributes "unspecified"
                        and specify it is part-of COURSE-ID,
```

```
            Specify NEXT-COURSE-STEP is next-step-after LESSON-STEP-ID
        If LESSON-FRAME-ID is entered
            If LESSON-FRAME-ID does not exist
                Add LESSON-FRAME for LESSON-FRAME-ID with attributes "unspecified",
            Specify LESSON-FRAME-ID is display-for LESSON-STEP-ID
        Display SCREEN-MSG of "Lesson Step Add Complete, Strike key to continue",
        Obtain keystroke from developer
END OF PROCESS


For each LESSON-STEP-TO-MODIFY:
        Display LESSON-STEP-PARAMETER-SCREEN with current values from LESSON-STEP
        for LESSON-STEP-ID,
        Obtain updates to NEXT-COURSE-STEP and LESSON-FRAME-ID from developer,
        Display an appropriate SCREEN-MSG and get input again if input is invalid,
        If entered NEXT-COURSE-STEP has changed
            If entered NEXT-COURSE-STEP is blank
                Specify the LESSON-STEP-ID leads-to no step,
            Else
                Specify the LESSON-STEP-ID leads-to entered NEXT-COURSE-STEP
        If entered LESSON-FRAME-ID has changed
            If entered LESSON-FRAME-ID is blank
                Specify there is no LESSON-FRAME that is display-for LESSON-STEP-ID
            Else
                Update LESSON-STEP-ID is displayed-with entered LESSON-FRAME-ID
        Display SCREEN-MSG of "Lesson Step Modify Complete. Press key to continue.",
        Obtain keystroke from the developer
END OF PROCESS


For each LESSON-STEP-TO-DISPLAY:
        Display LESSON-STEP-PARAMETER-SCREEN with the current values for
        LESSON-STEP-ID,
        Display SCREEN-MSG of Lesson Step Display Complete. Press key to continue",
        Obtain keystroke from developer
END OF PROCESS


For each COURSE-LESSON-STEPS-TO-LIST:
        Print LESSON-STEP-LISTING-HEADER,
        For all LESSON-STEPs which are part-of COURSE-ID:
            Sort by LESSON-STEP-ID,
            Print LESSON-STEP-LIST-LINE for each LESSON-STEP
        Display SCREEN-MSG of "Listing Complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each LESSON-STEP-TO-DELETE:
        Display SCREEEN-MSG of "Are you sure you want to delete?",
        Obtain DELETE-RESPONSE from developer'
        If DELETE-RESPONSE = "Y"
            Delete LESSON-STEP for LESSON-STEP-ID,
            Display SCREEN-MSG of "Lesson Step Delete Complete. Press key to continue.",
            Obtain keystroke from the developer
END OF PROCESS
```

```
For each ENTIRE-LESSON-STEP-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete?"
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                If there is a LESSON-FRAME specified as display-for LESSON-STEP-ID
                        If the LESSON-FRAME is not specified as the display-for any other
                        LESSON-STEP
                                Delete LESSON-FRAME
                Delete LESSON-STEP
                Display SCREEN-MSG indicating "Entire Lesson Step Delete Completed, Press
                key to continue.",
                Obtain keystroke from developer
END OF PROCESS


For each LESSON-FRAME-TO-ADD:
        Display LESSON-FRAME-DEVELOP-SCREEN for LESSON-FRAME-ID with blank
        GRAPHICS,
        Accept and process the developer's graphic directives as described in the general
        specification of Graphics Capabilities to develop new LESSON-FRAME
END OF PROCESS


For each LESSON-FRAME-TO-MODIFY:
        Display LESSON-FRAME-DEVELOP-SCREEN with current GRAPHICS for
        LESSON-FRAME-ID,
        Accept and process developer's graphic directives as described in the General
        Specifications for Graphics Capabilities to modify LESSON-FRAME
END OF PROCESS


For each LESSON-FRAME-ASSIGNMENT:
        Specify LESSON-FRAME-ID is display-for LESSON-STEP-ID,
        Display SCREEN-MSG of "Lesson Frame has been assigned to step, Press key to
        continue.",
        Obtain keystroke from developer
END OF PROCESS


For each LESSON-FRAME-TO-DISPLAY:
        Display LESSON-FRAME-DEVELOP-SCREEN with current GRAPHICS for
        LESSON-FRAME-ID,
        Display SCREEN-MSG of "Lesson Frame Display complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each LESSON-FRAME-LIST-REQ:
        Print LESSON-FRAME-LIST-HEADER,
        For all LESSON-FRAMEs that are display-for a LESSON-STEP which is part-of
        COURSE-ID:
                Sort them by LESSON-FRAME-ID,
                Print LESSON-FRAME-LIST-LINE for each
        Display SCREEN-MSG of "Lesson Frame List Complete. Press key to continue.",
        Obtain keystroke from operator
END OF PROCESS
```

```
For each LESSON-FRAME-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete",
        Obtain DELETE-RESPONSE from the developer,
        If DELETE-RESPONSE is "Yes"
                If LESSON-FRAME is-display-for only LESSON-STEPs that are-contained-in
                COURSE-ID or LESSON-FRAME is not display-for any LESSON-STEP
                        Delete LESSON-FRAME,
                Else
                        Remove that LESSON_FRAME is-display-for any LESSON-STEP that
                        is-contained-in COURSE-ID
                Display SCREEN-MSG of "Lesson Frame Delete Complete. Press key to
                continue.",
                Obtain keystroke from developer
END OF PROCESS


For each OPTION-STEP-TO-ADD:
        Display OPTION-PARAMETER-SCREEN for OPTION-STEP-ID with fields blank,
        Obtain OPTION-FRAME-ID from developer,
        If the input is invalid
                Display appropriate SCREEN-MSG and get input again,
        If OPTION-FRAME-ID does not exist
                Create OPTION-FRAME for OPTION-FRAME-ID with attributes "unspecified"
        Add OPTION-STEP for OPTION-STEP-ID,
        Specify OPTION-FRAME-ID is display-for OPTION-STEP-ID,
        Display SCREEN-MSG of "Option Step Add Complete. Press a key to continue",
        Obtain keystroke from the developer
END OF PROCESS.


For each OPTION-STEP-TO_MODIFY:
        Display OPTION-STEP-PARAMETER-SCREEN with current values for OPTION-STEP-ID,
        Obtain OPTION-FRAME-ID from developer,
        Display appropriate SCREEN-MSG if input value is invalid and get input again,
        If OPTION-FRAME-ID does not exist
                Create OPTION-FRAME for OPTION-FRAME-ID with attributes "unspecified"
        Specify OPTION-FRAME-ID is display-for OPTION-STEP-ID
        Display SCREEN-MSG of "Option Step Modify Complete, Press a key to continue",
        Obtain keystroke from the developer
END OF PROCESS.


For each OPTION-STEP-TO-DISPLAY:
        Display OPTION-STEP-PARAMETER-SCREEN with current values for OPTION-STEP-ID,
        Display SCREEN-MSG of "Option Step Display Complete. Press any key to continue."
        Obtain keystroke from the developer.
END OF PROCESS.


For each COURSE-OPTION-STEP-LIST-REQ:
        Print OPTION-STEP-LIST-HEADER,
        For all OPTION-STEPs that are part-of COURSE-ID:
                Print OPTION-STEP-LIST-LINE for each OPTION-STEP sorted by
                OPTION-STEP-ID,
```

```
        Display SCREEN-MSG of "List Complete. Press any key to continue."
        Obtain keystroke from developer.
END OF PROCESS.


For each OPTION-STEP-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete?",
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                Delete OPTION-CHOICEs that are specified for OPTION-STEP-ID,
                Delete OPTION-STEP for OPTION-STEP-ID,
                Display SCREEN-MSG of "Delete Complete. Press any key to continue."
                Obtain keystroke from developer,
END OF PROCESS.


For each ENTIRE-OPTION-STEP-TO-DELETE:
        Display SCREEN-MSG of "Are yoy sure you want to delete?",
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                Delete OPTION-CHOICEs specified for OPTION-STEP-ID,
                If a OPTION-FRAME which is display-for OPTION-STEP-ID exists, delete it if it's
                not display-for any other OPTION-STEP,
                Delete OPTION-STEP for OPTION-STEP-ID,
                Display SCREEN-MSG of "Delete Complete. Press key to continue",
                Obtain keystoke from developer.
END OF PROCESS


For each OPTION-FRAME-TO-ADD:
        Display OPTION-FRAME-DEVELOP-SCREEN with blank GRAPHICS,
        Accept and process graphics building directives as specified in the general
        specifications for graphic capabilities and associate graphics to OPTION-FRAME-ID.
END OF PROCESS.


For each OPTION-FRAME-TO-MODIFY:
        Display OPTION-FRAME-DEVELOP-SCREEN with current GRAPHICS for
        OPTION-FRAME-ID,
        Accept and process graphics building directives as specified in the General
        Specifications for Graphic Capabilities and update GRAPHICS for OPTION-FRAME-ID.
END OF PROCESS.


For each OPTION-FRAME-TO-ASSIGN-TO-STEP:
        Specify OPTION-FRAME-ID is display-for OPTION-STEP-ID,
        Display SCREEN-MSG of "Option Frame has been assigned to step. Press key to
        continue.",
        Obtain keystroke from developer
END OF PROCESS


For each OPTION-FRAME-TO-DISPLAY:
        Display OPTION-FRAME-DEVELOP-SCREEN with current GRAPHICS for
        OPTION-FRAME-ID,
```

```
        Display SCREEN-MSG of "Option Frame Display Complete. Press key to continue."
        Obtain keystroke from developer
END OF PROCESS.


For each OPTION-FRAME-LIST-REQ:
        Print OPTION-FRAME-LIST-HEADER,
        For all OPTION-FRAME that are display-for a LESSON-STEP which is part-of
        COURSE-ID:
                Sort by OPTION-FRAME-ID,
                Print OPTION-FRAME-LIST-LINE for each
        Display SCREEN-MSG of "Option Frame List Complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS.


For each OPTION-FRAME-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete?",
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Yes"
                If OPTION-FRAME-ID is display-for only OPTION-STEPs that are contained-in
                COURSE-ID or OPTION-FRAME-ID is not display-for any OPTION-STEP
                        Delete OPTION-FRAME and its assigned OPTION-CHOICEs
                Else
                        Remove that OPTION-FRAME is display-for any OPTION-STEP that is
                        contained-in COURSE-ID
                Display SCREEN-MSG of "Option Frame Delete Complete. Press key to
                continue.",
                Obtain keystroke from developer
END OF PROCESS.


For each OPTION-AREA-TO-ADD:
        If GRAPHICS for OPTION-FRAME which is display-for OPTION-STEP-ID is "unspecified"
                Display SCREEN-MSG of "Option Frame not yet specified. Option area can not
                be added. Press key to continue.",
                Obtain keystroke from developer,
                END OF PROCESS
        Determine unused OPTION-NBR for OPTION-CHOICEs assigned to OPTION-STEP-ID,
        Display OPTION-AREA-SCREEN with GRAPHICS for OPTION-FRAME-ID which is
        display-for OPTION-STEP-ID,
        Accept and process developer's graphic directives as specified in General Specification
        for Graphic Capabilities to develop OPTION-CHOICE for OPTION-NBR assigned to
        OPTION-STEP-ID,
                If an option area was not defined (exit with no save from graphic development)
                        END OF PROCESS
        Display SCREEN-MSG of "Please specify next course step."
        Obtain NEXT-COURSE-STEP from developer,
        Assign OPTION-CHOICE to the NEXT-COURSE-STEP,
        Display SCREEN-MSG of "Option Add Complete. Press key to continue."
        Obtain keystroke from developer
END OF PROCESS


For each OPTION-AREA-TO-MODIFY:
        For each OPTION-CHOICE assigned to OPTION-STEP-ID:
                Display OPTION-LIST-LINE with the NEXT-COURSE-STEP of the
                COURSE-STEP-ID which is next for the option
```

```
        Display prompts for OPTION-NBR and NEXT-COURSE-STEP
        Obtain OPTION-NBR and NEXT-COURSE-STEP from developer,
        Update assignment of OPTION-CHOICE for OPTION-NBR with NEXT-COURSE-STEP,
        Display SCREEN-MSG of "Option next step modify complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each OPTION-AREA-TO-DISPLAY:
        For each OPTION-CHOICE assigned to OPTION-STEP-ID:
                Display OPTION-LIST-LINE
        Display SCREEN-MSG of "Select Option Area",
        Obtain OPTION-NBR from developer,
        Display OPTION-DISPLAY-SCREEN with GRAPHICS for OPTION-FRAME-ID which is
        display-for OPTION-STEP-ID and the OPTION-AREA for OPTION-NBR shaded,
        Display SCREEN-MSG of "Option Area Display Complete. Press key to continue.",
        Obtain keystroke from developer
END OF PROCESS


For each OPTION-AREA-LIST-REQ:
        Print OPTION-LIST-HEADER,
        For each OPTION-CHOICE assigned to OPTION-STEP-ID:
                Print OPTION-LIST-LINE
        Display SCREEN-MSG of "Option List Complete. Press key to continue."
        Obtain keystroke from the developer
END OF PROCESS


For each OPTION-AREA-TO-DELETE:
        For each OPTION-CHOICE assigned to OPTION-STEP-ID:
                Display OPTION-LIST-LINE
        Display SCREEN-MSG of "Select Option Area",
        Obtain OPTION-NBR from developer,
        Display SCREEN-MSG of "Are you sure you want to delete?",
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE = "Y"
                Delete OPTION-CHOICE for OPTION-NBR which is assigned to
                OPTION-STEP-ID,
                Display SCREEN-MSG of "Option Delete Complete. Press key to continue.",
                Obtain keystroke from developer
END OF PROCESS


For each ALL-OPTION-AREAS-TO-DELETE:
        Display SCREEN-MSG of "Are you sure you want to delete?",
        Obtain DELETE-RESPONSE from developer,
        If DELETE-RESPONSE is "Yes"
                Delete all OPTION-CHOICEs assigned to OPTION-STEP-ID,
                Display SCREEN-MSG of "Option Area delete complete. Press key to continue."
                Obtain keystroke from developer
END OF PROCESS.


For each START-STEP-FOR-COURSE:
        If COURSE-ID is assigned a first step
                Remove the assignment
        Assign COURSE-STEP-ID as the first-step-of COURSE-ID,
```

Display SCREEN-MSG of "Start Step Assignment Complete. Press key to continue.",
Obtain keystroke from developer
END OF PROCESS

For each START-STEP-DISPLAY-REQ:
Display START-STEP-DISPLAY-LINE with COURSE-STEP-ID which is first-step-of COURSE-ID,
Display SCREEN-MSG of "Start Step Display Complete. Press key to continue.",
Obtain keystroke from developer
END OF PROCESS.

For each CONTENT-RPT-REQ:
Print CONTENT-RPT-HEADER,
For each QUESTION-STEP contained-in COURSE-ID:
Print QUESTION-STEP-CONTENT-RPT-LINE,
For each QUESTION-GROUP used-by a QUESTION-STEP that is contained-in COURSE-ID:
Print QUESTION-GROUP-CONTENT-RPT-LINE,
For each QUESTION which is part-of one of the above QUESTION-GROUPs:
Print QUESTION-CONTENT-RPT-LINE (DEFINED-FLAG indicates whether the GRAPHICS have been specified, ANSWER-COUNT is the number of ANSWERs specified for the QUESTION),
For each LESSON-STEP contained-in COURSE-ID:
Print LESSON-STEP-CONTENT-RPT-LINE,
For each LESSON-FRAME which is display-for a LESSON-STEP that is contained-in COURSE-ID:
Print LESSON-FRAME-CONTENT-RPT-LINE (DEFINED-FLAG indicates whether the GRAPHICS have been specified),
For each OPTION-STEP contained-in COURSE-ID:
Print OPTION-STEP-CONTENT-RPT-LINE,
For each OPTION-FRAME which is display-for OPTION-STEP that is contained-in COURSE-ID:
Print OPTION-FRAME-CONTENT-RPT-LINE (DEFINED-FLAG indicates whether the GRAPHICS have been specified)
(NOTE: For values that have not been specified, print question marks in the printed field instead of blanks)

END OF PROCESS.

For each SEQUENCE-RPT-REQ:
Starting with the first step of COURSE-ID follow the sequencing of steps in the course in order to identify which steps are included in the sequence (DETACHED-FLAG = "No") and which are not (DETACHED-FLAG = "Yes"),
Print SEQ-RPT-HEADER,
For each QUESTION-STEP contained-in COURSE-ID:
Print QUESTION-SEQ-RPT-LINE,
For each LESSON-STEP contained-in COURSE-ID:
Print LESSON-SEQ-RPT-LINE,
For each OPTION-STEP contained-in COURSE-ID:
Print OPTION-SEQ-RPT-LINE,
For each OPTION-CHOICE specified for OPTION-STEP:
Print OPTION-CHOICE-SEQ-RPT-LINE
END OF PROCESS.

GENERAL SPECIFICATIONS FOR GRAPHIC CAPABILITIES

For Question, Lesson frame and Option frame development, the following options for drawing are desirable. The options are displayed in a menu on a defined portion of the screen and the drawing is done on another portion. The developer chooses the option with the mouse. The developer is prompted for the required inputs of the option as described below. The input is validated to verify that the drawing is remaining in the drawing area. The display is updated as specified below.

TEXT	Input: mouse pressed to indicate the coordinates of the start of the text, text is typed followed by special termination function key. Display: the text as typed at the coordinates chosen, For a carriage return the text continues at the next line at the same horizontal coordinate as the first character.
LARGE TEXT	Same as for TEXT option with larger characters.
BOX	Input: mouse pressed at top left corner of box and the bottom right corner of the box. Display: a rectangle with the corners that were specified.
SMALL SQUARE	Input: mouse pressed at the top left corner. Display: a square of predefined "small" size displayed at the specified coordinates.
LARGE SQUARE	Same as SMALL SQUARE except larger.
CIRCLE	Input: mouse pressed at the coordinates of the center of the circle and at one of the points on the circle. Display: a circle as specified.
SMALL CIRCLE	Input: mouse pressed at the center of the circle. Display: a circle of predefined "small" size centered at the specified coordinates.
LARGE CIRCLE	Same as SMALL CIRCLE except larger circle is displayed.
LINE	Input: mouse pressed at endpoints of connected lines followed by pressing the mouse at coordinates outside of drawing area to indicate end of lines. Display: straight lines between the endpoints specified until completion is specified.
DRAW	Input: mouse pressed and held until done drawing. Display: a line following the mouse while it is pressed.
ERASE	Input: mouse pressed and held while moving over the portions of drawing to delete. Display: pixels are turned off in the path of the mouse while it is pressed.

CLEAR	Input: response to "Are you sure you want to clear?" Display: all pixels off (blank drawing).
READ STEP	Input: the COURSE-STEP-ID Validation: the COURSE-STEP-ID exists and has GRAPHICS specified. Display: the GRAPHICS specified for the COURSE-STEP-ID.
READ FILE	Input: the file name Validation: the file exists. Display: the GRAPHICS represented in the file (perhaps created by another DRAW program or written with the WRITE FILE option)
WRITE FILE	Input: the file name Validation: the file does not already exist. Display: None
EXIT	Input: response to "do you want to save?" Display: None (control is transfered back to the initiator of the graphics processing)

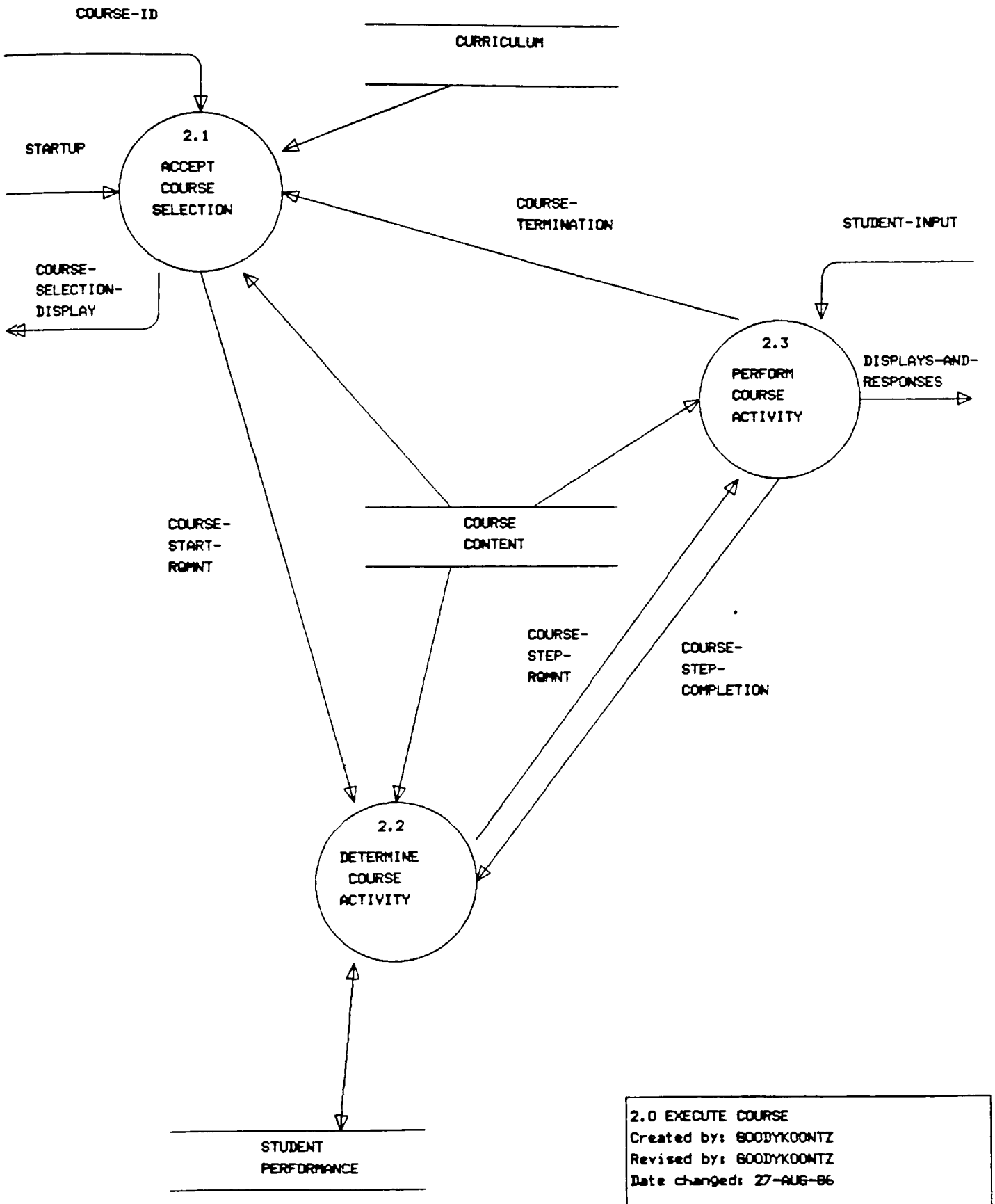
For Answer Area specification and Option Area specification, the above drawing options: BOX, SMALL SQUARE, LARGE SQUARE, CIRCLE, SMALL CIRCLE, LARGE CIRCLE, LINE, DRAW and ERASE are provided along with the following options.

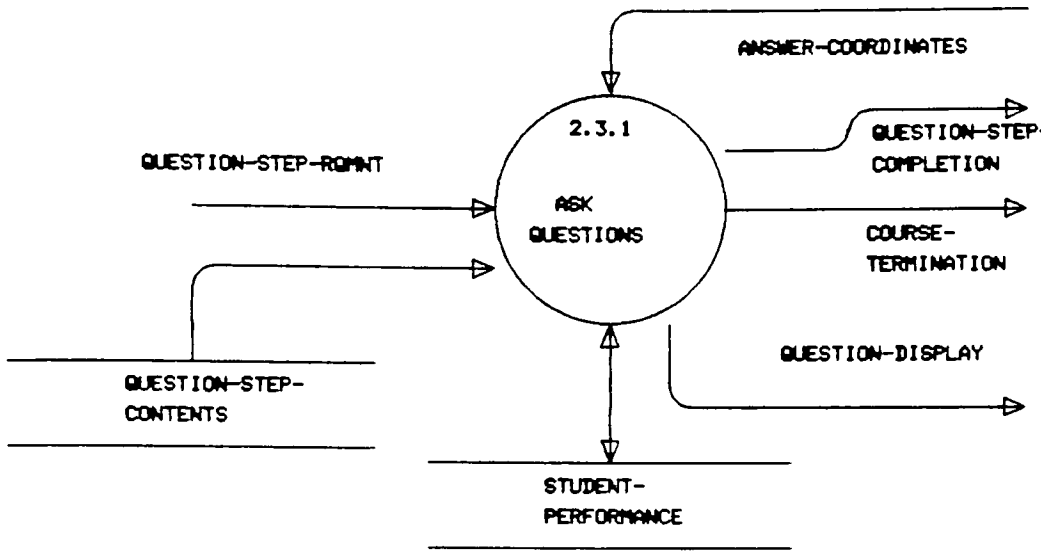
--Area specification options--

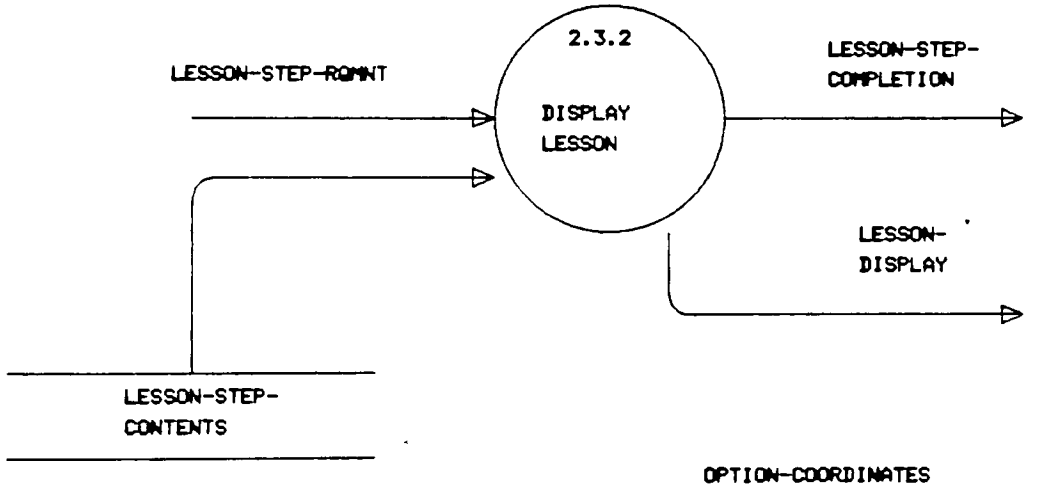
ANSWER BOX	Input: same as BOX Display: a shaded in rectangle. Defined Area: the entire rectangle (sides and inside)
ANSWER CIRCLE	Input: same as CIRCLE Display: a shaded in circle. Defined Area: the entire circle (edge and inside).
ANSWER AREA	Input: mouse presses at point within bounded area. Display: shade in the area. Defined Area: within the boundaries of the area but not the boundary.

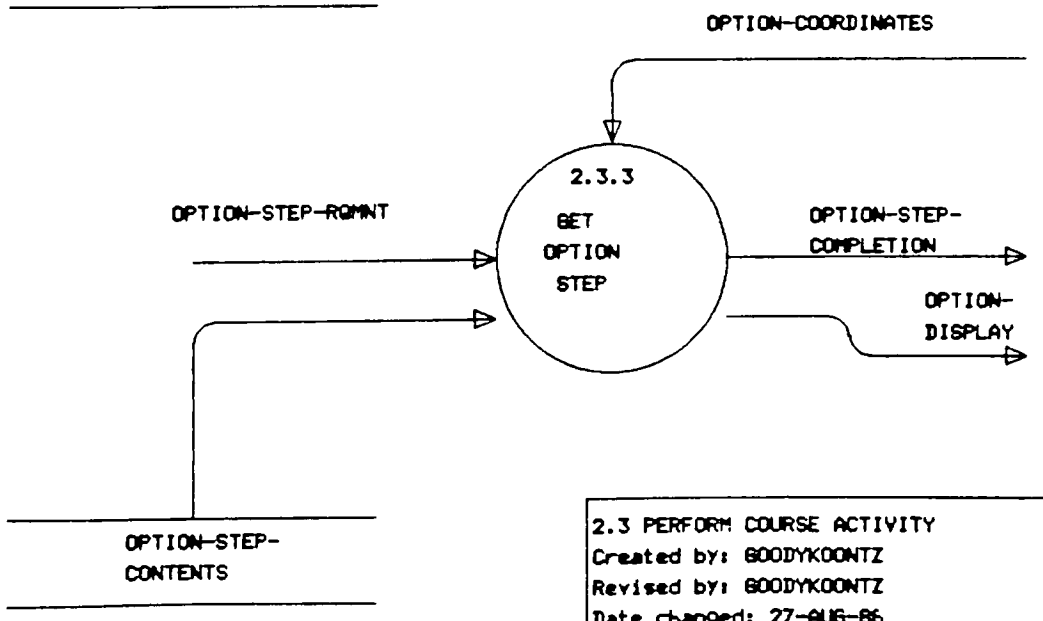
--Misc Options--

CLEAR	Input: response to "Do you want to clear area specification activity?" Display: If response = Yes, return the display to its appearance when the function was entered.
EXIT	Input: response to "do you want to save area specification?" Display: none (control is returned to initiator of Graphics function).

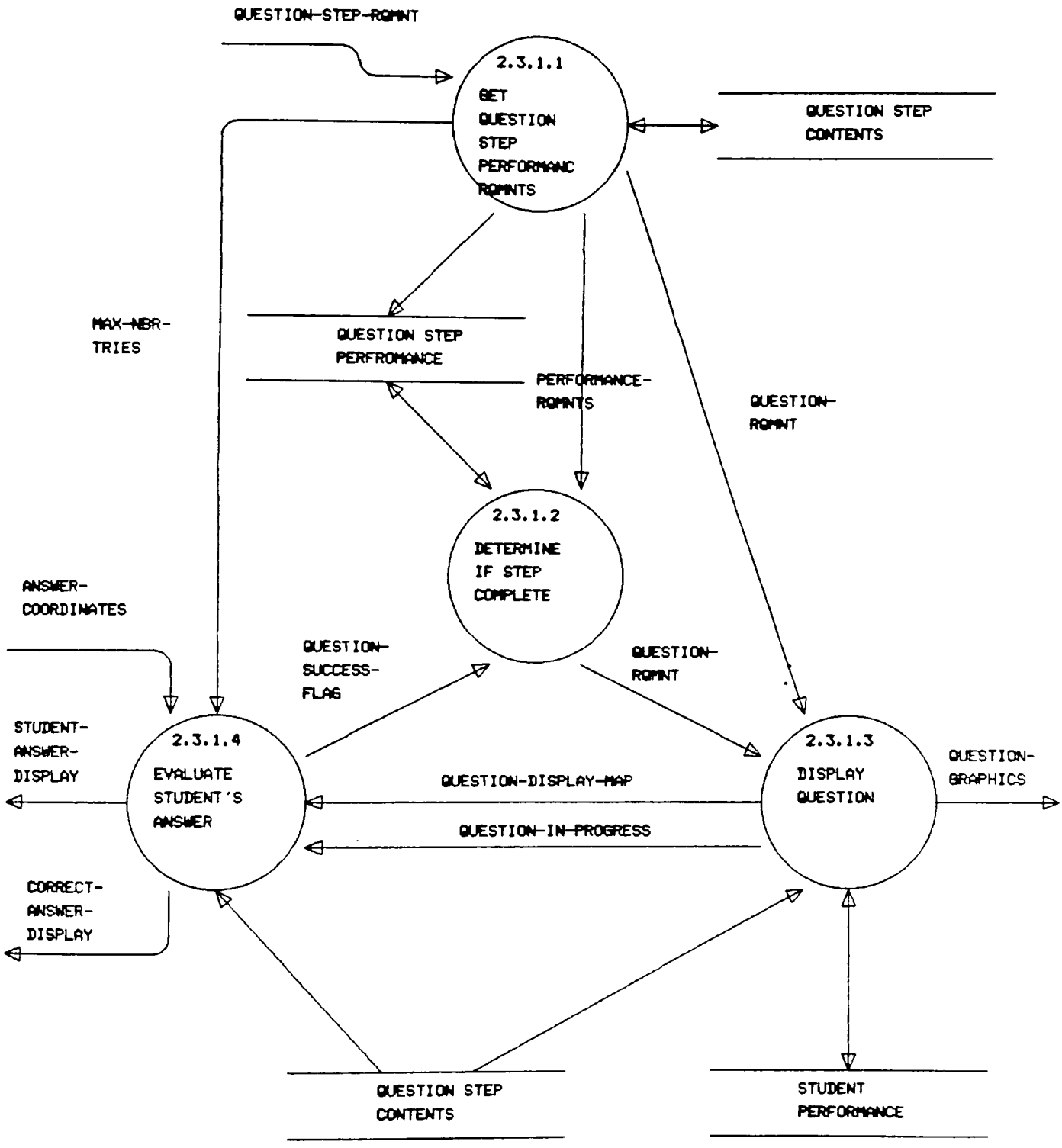

COURSE-ID
CURRICULUM
STARTUP
2.1
ACCEPT
COURSE
SELECTION
COURSE-
TERMINATION
STUDENT-INPUT
COURSE-
SELECTION-
DISPLAY
2.3
PERFORM
COURSE
ACTIVITY
DISPLAYS-AND-
RESPONSES
COURSE-
START-
RQMNT
COURSE
CONTENT
COURSE-
STEP-
RQMNT
COURSE-
STEP-
COMPLETION
2.2
DETERMINE
COURSE
ACTIVITY
STUDENT
PERFORMANCE
2.0 EXECUTE COURSE
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 27-AUG-86







2.3 PERFORM COURSE ACTIVITY
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 27-AUG-86



2.3.1 ASK QUESTIONS
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 27-AUG-86

2.0 EXECUTE COURSE.

2.1 ACCEPT COURSE SELECTION.
For each STARTUP or COURSE-TERMINATION:
Display COURSE-SELECTION-DISPLAY with all COURSE-IDs that are accessible-to STUDENT-ID and do not have a COURSE-COMPLETION specified for STUDENT-ID,
Display DIRECTIVE-MSG of "Enter Course ID",
Obtain COURSE-ID from student,
If abort was pressed
END OF PROCESS
If the COURSE-ID is not accessible to STUDENT-ID
Display INFORMATIONAL-MSG of "Course not available." and go get the input again.
Create COURSE-START-RQMNT
END OF PROCESS

2.2 DETERMINE COURSE ACTIVITY.
For each COURSE-START-RQMNT or COURSE-STEP-COMPLETION:
If triggered by COURSE-START-RQMNT:
Determine CURRENT-STEP in COURSE-ID for STUDENT-ID,
If one does not exist (student has not yet accessed course)
Determine the COURSE-STEP-ID which is first-step-of COURSE-ID
If triggered by QUESTION-STEP-COMPLETION:
If STEP-COMPLETION-STATUS is "Success"
Determine the COURSE-STEP-ID which is next on success
Else
Determine the COURSE-STEP-ID which is next on failure
If triggered by LESSON-STEP-COMPLETION:
Determine the COURSE-STEP-ID which is next
If triggered by OPTION-STEP-COMPLETION:
the next COURSE-STEP-ID is NEXT-STEP-ID in COURSE-STEP-COMPLETION
If the determined next COURSE-STEP-ID is "END-SUCCESSFUL" or "END-UNSUCCESSFUL"
Specify a COURSE-COMPLETION of COURSE-ID by STUDENT-ID,
Delete CURRENT-STEP in COURSE-ID for STUDENT-ID
Else
Update CURRENT-STEP in COURSE-ID for STUDENT-ID to the determined COURSE-STEP-ID,
Create COURSE-STEP-RQMNT for the determined COURSE-STEP-ID

2.3 PERFORM COURSE ACTIVITY.

2.3.1 ASK QUESTIONS

2.3.1.1 GET QUESTION STEP PERFORMANCE RQMNTS.
For each QUESTION-STEP-RQMNT:
Get QUESTION-STEP for QUESTION-STEP-ID,
Initialize QUESTION-STEP-PERFORMANCE (NBR-QUEST-ANSWERED = 0, NBR-CORRECT-ANSWERS = 0, PERFORMANCE-QUEUE emptied),
Create MAX-NBR-TRIES,
Create PERFORMANCE-RQMNTS with values from QUESTION-STEP,
Create QUESTION-RQMNT for QUESTION-GROUP-ID in QUESTION-STEP
END OF PROCESS

2.3.1.2 DETERMINE IF QUESTION STEP IS COMPLETE.

For each QUESTION-SUCCESS-FLAG:

Update QUESTION-STEP-PERFORMANCE as follows

Increment NBR-QUEST-ANSWERED,

If PERFORMANCE-QUEUE has NBR-LAST-QUEST entries

Take one entry from beginning of queue,

If it was "SUCCESS"

Decrement NBR-CORRECT-ANSWERS

Add QUESTION-SUCCESS-FLAG to end of PERFORMANCE-QUEUE

If QUESTION-SUCCESS-FLAG is "Success"

Increment NBR-CORRECT-ANSWERS

If NBR-QUEST-ANSWERED >= MIN-QUEST-ASKED and NBR-CORRECT-ANSWERS >= NBR-CORRECT-ANSWERS-REQUIRED

Create QUESTION-STEP-COMPLETION with STEP-COMPLETION-STATUS of "Success"

Else

If NBR-QUEST-ANSWERED = MAX-QUEST-ASKED

Create QUESTION-STEP-COMPLETION with STEP-COMPLETION-STATUS of "Failure"

Else

Create QUESTION-RQMNT

END OF PROCESS

2.3.1.3 DISPLAY QUESTION.

For each QUESTION-RQMNT:

In a random fashion find a QUESTION-ID that is included-in QUESTION-GROUP-ID that STUDENT-ID has not already been asked (QUESTION-INSTANCE does not exist),

Specify that STUDENT-ID has been asked the found QUESTION-ID (add QUESTION-INSTANCE),

Display GRAPHICS for QUESTION-ID and create QUESTION-DISPLAY-MAP,

Display blank INFORMATIONAL-MSG,

Display DIRECTIVE-MSG of "Waiting for answer",

Create QUESTION-IN-PROGRESS

END OF PROCESS

2.3.1.4 EVALUATE STUDENT'S ANSWER.

For each QUESTION-IN-PROGRESS:

Initialize NBR-OF-TRIES to 0,

Obtain ANSWER-COORDINATES from student,

If abort key pressed

Create COURSE-TERMINATION

END OF PROCESS

IF ANSWER-COORDINATES are not in the Question Display Area

Display DIRECTIVE-MSG of "Choose point within question area" and go back and get input again,

Increment NBR-OF-TRIES,

Refresh display with QUESTION-DISPLAY-MAP,

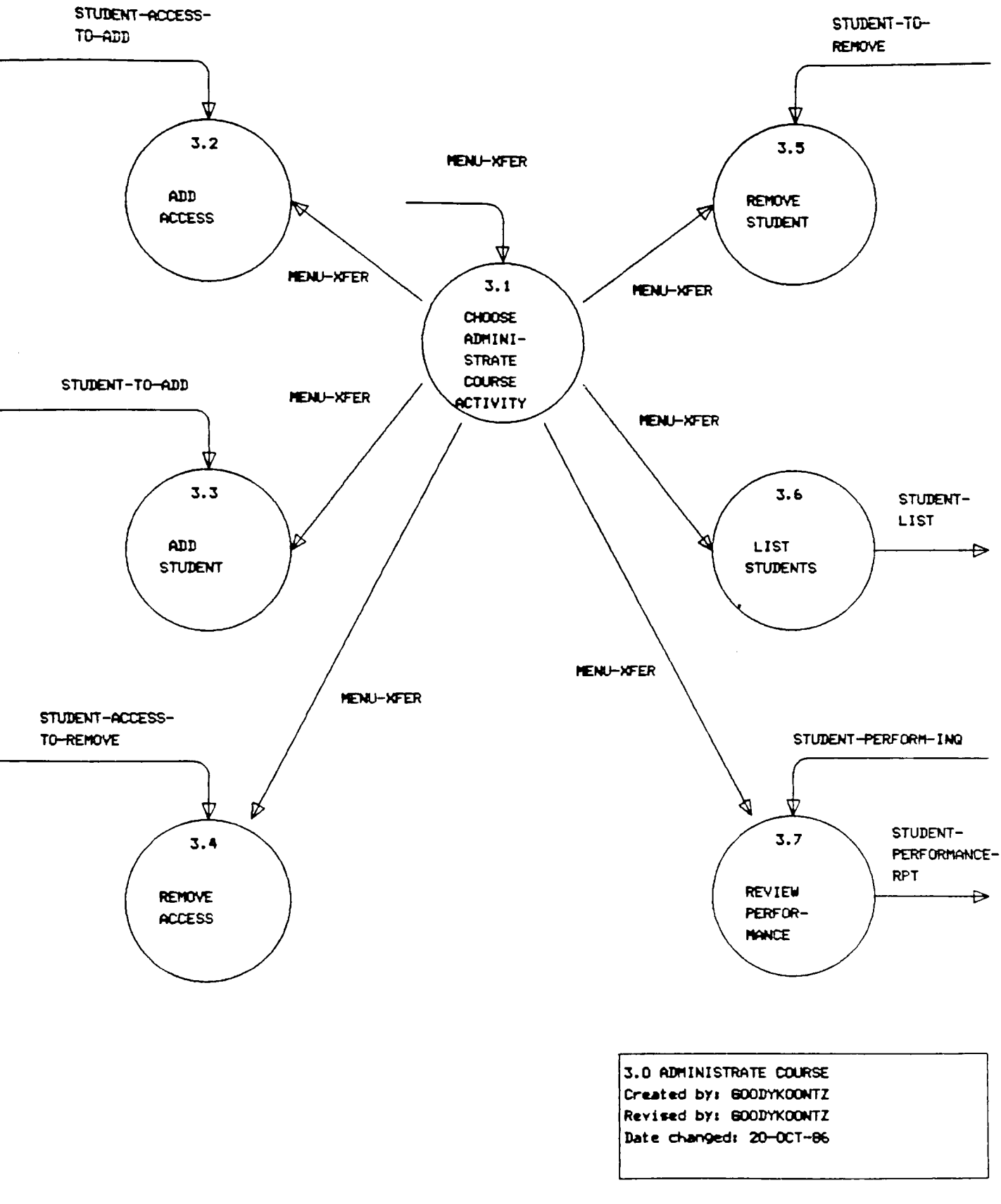
Display blinking point at ANSWER-COORDINATES,

Determine which ANSWER-AREA which is designated for QUESTION-ID contains ANSWER-COORDINATES,

Display INFORMATIONAL-MSG with the ANSWER-RESPONSE specified for the found ANSWER-AREA (if no ANSWER-AREA was found use the specified "DEFAULT" response, if no default response is specified for the question then display INFORMATIONAL-MSG of "Wrong answer")
If ANSWER-COORDINATES were in an ANSWER-AREA with ANSWER-TYPE = "RIGHT"
Display DIRECTIVE-MSG of "Press any key to continue",
Obtain keystroke from student,
Create QUESTION-SUCCESS-FLAG of "Success"
Else
If NBR-OF-TRIES >= MAX-NBR-OF-TRIES
Display DIRECTIVE-MSG of "Press any key for correct answer",
Obtain keystroke from student,
Refresh display from QUESTION-DISPLAY-MAP,
Display in blinking the ANSWER-AREAs for QUESTION-ID that have ANSWER-TYPE of "Right",
Display INFORMATIONAL-MSG of "Correct Answer is displayed",
Display DIRECTIVE-MSG of "Press any key to continue",
Obtain keystroke from student
Create QUESTION-SUCESS-FLAG of "Failure"
Else
Go back and obtain ANSWER-COORDINATES
END OF PROCESS

2.3.2 DISPLAY LESSON.
For each LESSON-STEP-RQMNT:
Display the GRAPHICS for LESSON-FRAME for LESSON-STEP-ID,
Display blank INFORMATIONAL-MSG,
Display DIRECTIVE-MSG of "Press any key to continue",
Obtain keystroke from student,
If abort pressed
Create COURSE-TERMINATION
END OF PROCESS
Create LESSON-STEP-COMPLETION
END OF PROCESS

2.3.3 GET OPTION CHOICE.
For each OPTION-STEP-RQMNT:
Display the GRAPHICS for the OPTION-FRAME for OPTION-STEP-ID,
Display DIRECTIVE-MSG of "Choose option",
Obtain OPTION-COORDINATES from student,
If abort was pressed
Create COURSE-TERMINATION
END OF PROCESS
If OPTION-COORDINATES are not in any OPTION-AREAs specified for OPTION-STEP-ID
Display DIRECTIVE-MSG of "invalid option" and go back to get input again,
Find OPTION-CHOICE which has OPTION-AREA that the OPTION-COORDINATES are within,
Create OPTION-STEP-COMPLETION containing the NEXT-STEP-ID for the OPTION-CHOICE
END OF PROCESS



3.0 ADMINISTRATE COURSE
Created by: GOODYKOONTZ
Revised by: GOODYKOONTZ
Date changed: 20-OCT-86

3.0 ADMINISTRATE COURSE.

3.1 CHOOSE ADMINISTRATE COURSE ACTIVITY.
For each MENU-XFER:
Display MAIN-MENU for Couse Administration,
Obtain MENU-CHOICE from administrator,
Transfer control to the choosen function (control to be returned to the main menu when a special function key is depressed)

3.2 ADD ACCESS.
For each MENU-XFER:
Display prompts and obtain STUDENT-ID and COURSE-ID from administrator,
Display an appropriate SCREEN-MSG and go back and get the input again for the following:
STUDENT-ID does not exist,
COURSE-ID does not exist,
STUDENT-ID already has access to COURSE-ID
Specify STUDENT-ID has access-to COURSE-ID,
Display SCREEN-MSG of "Activity complete. Press key to continue."
Obtain keystroke from administrator
END OF PROCESS.

3.3 ADD STUDENT.
For each MENU-XFER:
Display prompts and obtain STUDENT-ID and STUDENT-NAME from administrator,
Display appropriate SCREEN-MSG and go back to get input for the following:
STUDENT-ID already exists
Add STUDENT for STUDENT-ID,
Display "Activity Complete. Press key to continue.",
Obtain keystroke from administrator
END OF PROCESS.

3.4 REMOVE ACCESS.
For each MENU-XFER:
Display prompts and obtain STUDENT-ID and COURSE-ID from administrator,
Display appropriate SCREEN-MSG and go back to get input for the following:
STUDENT-ID doesn't exist,
COURSE-ID doesn't exist
If COURSE-COMPLETION of COURSE-ID by STUDENT-ID is not specifed
Display SCREEN-MSG of "Course is not completed by student. Do you still want to remove?"
Obtain CONFIRM-RESPONSE from administrator,
If CONFIRM-RESPONSE is "No"
END OF PROCESS
Remove specification of COURSE-ID accessible-to STUDENT-ID
Display SCREEN-MSG of "Activity Complete. Press key to continue"
Obtain keystroke from administrator
END OF PROCESS.

```
3.5 REMOVE STUDENT.
    For each MENU-XFER:
        Display prompts and obtain STUDENT-ID and STUDENT-NAME from administrator,
        Display appropriate SCREEN-MSG and go back to get input for the following:
            STUDENT-ID does not exist,
            STUDENT-ID exists but entered STUDENT-NAME does not match
        Delete STUDENT for STUDENT-ID,
        Display SCREEN-MSG of "Activity complete. Press key to continue",
        Obtain keystroke form administrator
    END OF PROCESS.

3.6 LIST STUDENTS.
    For each MENU-XFER:
        Print STUDENT-LIST-HEADER,
        For each STUDENT:
            Print STUDENT-LIST-LINE,
    END OF PROCESS.

3.7 REVIEW PERFORMANCE.
    For each MENU-XFER:
        Display prompt and obtain STUDENT-ID from administrator,
        If STUDENT-ID does not exist
            Display SCREEN-MSG of "Student does not exist"
        Else
            Print PERFORMANCE-HEADER-LINE,
            For each COURSE-COMPLETION specified for STUDENT-ID:
                Print COMPLETE-COURSE-LINE,
            For each CURRENT-STEP specified for STUDENT-ID:
                Print INPROGRESS-COURSE-LINE,
            For each course not yet started by STUDENT-ID:
                Print NOT-STARTED-COURSE-LINE
    END OF PROCESS
```

ADD-ANSWER-SCREEN	=	QUESTION-ID + GRAPHICS
ALL-OPTION-AREAS-TO-DELETE	=	COURSE-ID + OPTION-STEP-ID
ANSWER	=	ANSWER-NBR + QUESTION-REF + ANSWER-TYPE + ANSWER-AREA + ANSWER-RESPONSE
ANSWER-AREA	=	"DEFAULT" | AREA
ANSWER-AREA-SCREEN	=	QUESTION-ID + GRAPHICS + ANSWER-AREA
ANSWER-AREA-TO-ADD	=	QUESTION-ID + ANSWER-TYPE
ANSWER-AREA-TO-DELETE	=	QUESTION-ID
ANSWER-AREA-TO-DISPLAY	=	QUESTION-ID
ANSWER-AREA-TO-LIST	=	QUESTION-ID
ANSWER-AREA-TO-MODIFY	=	QUESTION-ID
ANSWER-COORDINATES	=	XCOORDINATE + YCOORDINATE
ANSWER-DEV-ACTIVITY-DATA	=	ANSWER-AREA-TO-ADD + ANSWER-AREA-TO-MODIFY + ANSWER-AREA-TO-DISPLAY + ANSWER-AREA-TO-LIST + ANSWER-AREA-TO-DELETE
ANSWER-DISPLAY-SCREEN	=	QUESTION-ID + ANSWER-NBR + GRAPHICS + ANSWER-AREA
ANSWER-LIST-HEADER	=	QUESTION-ID
ANSWER-LIST-LINE	=	ANSWER-NBR + ANSWER-RESPONSE
ANSWER-NBR	=	Identifier for an answer to a specific question
ANSWER-RESPONSE	=	Response to be displayed to student
ANSWER-TYPE	=	"RIGHT" | "WRONG"
AREA	=	Description of subset of pixels within display area
AREA-TYPE	=	"D" | "A"
COMPLETE-COURSE-LINE	=	COURSE-ID + COMPLETION-STATUS + DATE + TIME

COMPLETION-STATUS = "SUCCESS" | "FAILURE"

COMPLETION-STEP-ID = COURSE-STEP-ID

CONTENT-RPT = CONTENT-RPT-HEADER +
{QUESTION-STEP-CONTENT-RPT-LINE} +
{QUESTION-GROUP-CONTETN-RPT-LINE} +
{QUESTION-CONTENT-RPT-LINE} +
{LESSON-STEP-CONTENT-RPT-LINE} +
{LESSON-FRAME-CONTENT-RPT-LINE} +
{OPTION-STEP-CONTENT-RPT-LINE} +
{OPTION-FRAME-CONTENT-RPT-LINE}

CONTENT-RPT-HEADER = COURSE-ID + DATE + TIME

CONTENT-RPT-REQ = COURSE-ID

CORRECT-ANSWER-DISPLAY = Display of correct ANSWER-AREA for a question

COURSE = COURSE-ID + COURSE-TITLE

COURSE-ACTIVITY-DATA = COURSE-LIST-REQ | COURSE-TO-DEVELOP |
EXIT-REQ

COURSE-ACTIVITY-INPUT = MENU-CHOICE + COURSE-ID

COURSE-COMPLETION = STUDENT-REF + COURSE-REF + DATE + TIME +
COMPLETION-STATUS + COMPLETION-STEP-ID

COURSE-COMPONENTS = See Entity Relation Diagram

COURSE-CONTENT = COURSE-COMPONENTS +
QUESTION-STEP-CONTENTS +
LESSON-STEP-CONTENTS +
OPTION-STEP-CONTENTS

COURSE-DEV-ACTIVITY-OPERATOR-INPUT = See Table 1-3

COURSE-DEVELOPMENT-OPERATOR-INPUT = {QUESTION-STEP-PARAMETERS} +
{SUCCESS-COURSE-STEP} +
{FAILURE-COURSE-STEP} + {DELETE-RESPONSE}
+ {ANSWER-RESPONSE} + {AREA-TYPE} +
{ANSWER-NBR} + {NEXT-COURSE-STEP} +
{LESSON-FRAME-ID} + {OPTION-FRAME-ID} +
{OPTION-NBR} + {GRAPHIC-INPUT}

COURSE-LESSON-STEPS-TO-LIST = COURSE-ID

COURSE-LIST-HEADER = DATE + TIME

COURSE-LIST-LINE = COURSE-ID + COURSE-TITLE

COURSE-LIST-REQ	=	Trigger
COURSE-LISTING	=	COURSE-LIST-HEADER + {COURSE-LIST-LINE}
COURSE-OPTION-STEP-LIST-REQ	=	COURSE-ID
COURSE-QUESTION-STEPS-TO-LIST	=	COURSE-ID
COURSE-SELECTION-DISPLAY	=	{COURSE-ID + COURSE-TITLE}
COURSE-START-RQMNT	=	COURSE-ID + STUDENT-ID
COURSE-STEP	=	LESSON-STEP | OPTION-STEP | QUESTION-STEP | "END-SUCCESS" | "END-FAILURE"
COURSE-STEP-COMPLETION	=	QUESTION-STEP-COMPLETION | LESSON-STEP-COMPLETION | OPTION-STEP-COMPLETION
COURSE-STEP-ID	=	STEP-TYPE + STEP-ID
COURSE-STEP-RQMNT	=	LESSON-STEP-RQMNT | QUESTION-STEP-RQMNT | OPTION-STEP-RQMNT
COURSE-TERMINATION	=	STUDENT-ID
COURSE-TO-DEVELOP	=	COURSE-ID
CURRENT-COURSE-STEP	=	COURSE-STEP-ID
CURRENT-STEP	=	COURSE-STEP-REF + STUDENT-REF + DATE + TIME
CURRICULUM	=	See entity Relation Diagram
DELETE-RESPONSE	=	Yes/No response for deletion confirmation
DEV-LISTINGS-AND-REPORTS	=	{QUESTION-STEP-LISTING} + {QUESTION-GROUP-LISTING} + {QUESTION-LISTING} + {ANSWER-LIST} + {LESSON-STEP-LISTING} + {LESSON-FRAME-LIST} + {OPTION-STEP-LIST} + {OPTION-FRAME-LIST} + {OPTION-LIST} + {CONTENT-RPT} + {SEQUENCE-RPT}
DEVELOPMENT-ACTIVITY-DATA	=	QUEST-DEV-ACTIVITY-DATA | QUEST-GROUP-DEV-ACTIVITY-DATA | QUEST-STEP-DEV-ACTIVITY-DATA | ANSWER-DEV-ACTIVITY-DATA | LESSON-STEP-DEV-ACTIVITY-DATA | LESSON-FRAME-DEV-ACTIVITY-DATA | OPTION-STEP-DEV-ACTIVITY-DATA |

		OPTION-FRAME-DEV-ACTIVITY-DATA | OPTION-AREA-DEV-ACTIVITY-DATA | START-STEP-DEV-ACTIVITY-DATA | REPORT-DEV-ACTIVITY-DATA
DIRECTIVE-MSG	=	Text giving direction that is to displayed on a dedicated portion of the display
DISPLAYS-AND-RESPONSES	=	{OPTION-DISPLAY} + {LESSON-DISPLAY} + {QUESTION-DISPLAY} + {DIRECTIVE-MSG} + {INFORMATIONAL-MSG}
ENTIRE-LESSON-STEP-TO-DELETE	=	COURSE-ID + LESSON-STEP-ID
ENTIRE-OPTION-STEP-TO-DELETE	=	COURSE-ID + OPTION-STEP-ID
ENTIRE-QUESTION-GROUP-TO-DELETE	=	QUESTION-GROUP-ID + COURSE-ID
ENTIRE-QUESTION-STEP-TO-DELETE	=	COURSE-ID + QUESTION-STEP-ID
FAILURE-COURSE-STEP	=	COURSE-STEP-ID
FILENAME	=	Name of a file
GRAPHIC-DISPLAY	=	Lines, circles, boxes, text displayed on screen
GRAPHIC-INPUT	=	{MOUSE-INPUT} + {FILENAME} + {COURSE-STEP-ID} + {YES-NO-RESPONSE}
GRAPHICS	=	Description of illuminated pixel for Lesson, Question and Option Displays
INFORMATIONAL-MSG	=	Informational text to be displayed on a dedicated portion of the display
INPROGRESS-COURSE-LINE	=	COURSE-ID + CURRENT-COURSE-STEP + DATE + TIME
LESSON-DISPLAY	=	Display of GRAPHICS for LESSON-FRAME
LESSON-FRAME	=	LESSON-FRAME-ID + GRAPHICS
LESSON-FRAME-ASSIGNMENT	=	COURSE-ID + LESSON-FRAME-ID + LESSON-STEP-ID
LESSON-FRAME-CONTENT-RPT-LINE	=	LESSON-FRAME-ID + DEFINED-FLAG
LESSON-FRAME-DEVELOP-SCREEN	=	LESSON-FRAME-ID + GRAPHICS
LESSON-FRAME-DEV-	=	LESSON-FRAME-TO-ADD |

ACTIVITY-DATA		LESSON-FRAME-TO-MODIFY | LESSON-FRAME-ASSIGNMENT | LESSON-FRAME-TO-DISPLAY | LESSON-FRAME-LIST-REQ | LESSON-FRAME-TO-DELETE
LESSON-FRAME-ID	=	Unique identifier for Lesson Frame
LESSON-FRAME-LIST	=	LESSON-FRAME-LIST-HEADER + {LESSON-FRAME-LIST-LINE}
LESSON-FRAME-LIST-HEADER	=	COURSE-ID + DATE + TIME
LESSON-FRAME-LIST-LINE	=	LESSON-FRAME-ID + LESSON-STEP-ID
LESSON-FRAME-LIST-REQ	=	COURSE-ID
LESSON-FRAME-TO-ADD	=	LESSON-FRAME-ID
LESSON-FRAME-TO-DELETE	=	LESSON-FRAME-ID
LESSON-FRAME-TO-DISPLAY	=	LESSON-FRAME-ID
LESSON-FRAME-TO-MODIFY	=	LESSON-FRAME-ID
LESSON-SEQ-RPT-LINE	=	LESSON-STEP-ID + NEXT-COURSE-STEP + DETACHED-FLAG
LESSON-STEP	=	LESSON-STEP-ID + COURSE-REF
LESSON-STEP-COMPLETION	=	LESSON-STEP-ID + STUDENT-ID
LESSON-STEP-CONTENT- RPT-LINE	=	LESSON-STEP-ID + LESSON-FRAME-ID
LESSON-STEP-CONTENTS	=	See Entity Relation Diagram
LESSON-STEP-DEV- ACTIVITY-DATA	=	LESSON-STEP-TO-ADD | LESSON-STEP-TO-MODIFY | LESSON-STEP-TO-DISPLAY | COURSE-LESSON-STEPS-LIST-REQ | LESSON-STEP-TO-DELETE | ENTIRE-LESSON-STEP-TO-DELETE
LESSON-STEP-ID	=	COURSE-STEP-ID where STEP-TYPE = "LESSON"
LESSON-STEP-LISTING	=	LESSON-STEP-LISTING-HEADER + {LESSON-STEP-LIST-LINE}
LESSON-STEP-LISTING-HEADER	=	COURSE-ID + DATE + TIME
LESSON-STEP-LIST-LINE	=	LESSON-STEP-ID + NEXT-COURSE-STEP + LESSON-FRAME-ID

LESSON-STEP-PARAMETER-SCREEN	=	LESSON-STEP-ID + NEXT-COURSE-STEP + LESSON-FRAME-ID
LESSON-STEP-RQMNT	=	COURSE-ID + LESSON-STEP-ID + STUDENT-ID
LESSON-STEP-TO-ADD	=	COURSE-ID + LESSON-STEP-ID
LESSON-STEP-TO-DELETE	=	COURSE-ID + LESSON-STEP-ID
LESSON-STEP-TO-DISPLAY	=	COURSE-ID + LESSON-STEP-ID
LESSON-STEP-TO-MODIFY	=	COURSE-ID + LESSON-STEP-ID
MAX-NBR-TRIES	=	Number of chances given to answer a question
MAX-QUEST-ASKED	=	Maximum number of questions asked
MENU-CHOICE	=	Choice of activity from menu
MENU-XFER	=	Trigger
MIN-QUEST-ASKED	=	Minimum number of questions asked
MOUSE-INPUT	=	Coordinates identified when pressing mouse
NEW-COURSE-INFO	=	COURSE-TITLE
NEW-COURSE-TO-DEVELOP	=	COURSE-ID
NEXT-COURSE-STEP	=	COURSE-STEP-ID
NBR-CORRECT-ANSWERS	=	Number of questions answered correctly
NBR-CORRECT-ANSWERS-REQUIRED	=	Number of questions that must be answered correctly for success
NBR-OF-TRIES	=	Number of tries at answering a question
NBR-LAST-QUEST	=	The number of last questions that are considered in determining success
NOT-STARTED-COURSE-LINE	=	COURSE-ID
OPTION-AREA	=	AREA
OPTION-AREA-DEV ACTIVITY-DATA	=	OPTION-AREA-TO-ADD | OPTION-AREA-TO-MODIFY | OPTION-AREA-TO-DISPLAY | OPTION-AREA-LIST-REQ | OPTION-AREA-TO-DELETE | ALL-OPTION-AREAS-TO-DELETE
OPTION-AREA-LIST-REQ	=	COURSE-ID + OPTION-STEP-ID

OPTION-AREA-SCREEN	=	GRAPHICS + OPTION-FRAME-ID + OPTION-NBR
OPTION-AREA-TO-ADD	=	COURSE-ID + OPTION-STEP-ID
OPTION-AREA-TO-DELETE	=	COURSE-ID + OPTION-STEP-ID
OPTION-AREA-TO-DISPLAY	=	COURSE-ID + OPTION-STEP-ID
OPTION-AREA-TO-MODIFY	=	COURSE-ID + OPTION-STEP-ID
OPTION-CHOICE	=	OPTION-NBR + OPTION-STEP-REF + OPTION-AREA
OPTION-DISPLAY	=	Display of GRAPHICS for OPTION-FRAME
OPTION-DISPLAY-SCREEN	=	GRAPHICS + AREA
OPTION-COORDINATES	=	XCOORDINATE + YCOORDINATE
OPTION-FRAME	=	OPTION-FRAME-ID + GRAPHICS
OPTION-FRAME-DEVELOP-SCREEN	=	OPTION-FRAME-ID + GRAPHICS
OPTION-FRAME-CONTENT-RPT-LINE	=	OPTION-FRAME-ID + DEFINED-FLAG
OPTION-FRAME-DEV-ACTIVITY-DATA	=	OPTION-FRAME-TO-ADD | OPTION-FRAME-TO-MODIFY | OPTION-FRAME-TO-ASSIGN-TO-STEP | OPTION-FRAME-TO-DISPLAY | OPTION-FRAME-LIST-REQ | OPTION-FRAME-TO-DELETE
OPTION-FRAME-ID	=	Unique identifier of Option Frame
OPTION-FRAME-LIST	=	OPTION-FRAME-LIST-HEADER + {OPTION-FRAME-LIST-LINE}
OPTION-FRAME-LIST-HEADER	=	COURSE-ID
OPTION-FRAME-LIST-LINE	=	OPTION-FRAME-ID + OPTION-STEP-ID
OPTION-FRAME-LIST-REQ	=	COURSE-ID
OPTION-FRAME-TO-ADD	=	COURSE-ID + OPTION-FRAME-ID
OPTION-FRAME-TO-ASSIGN-TO-STEP	=	COURSE-ID + OPTION-STEP-ID + OPTION-FRAME-ID
OPTION-FRAME-TO-DELETE	=	COURSE-ID + OPTION-FRAME-ID
OPTION-FRAME-TO-DISPLAY	=	COURSE-ID + OPTION-FRAME-ID

OPTION-FRAME-TO-MODIFY	=	COURSE-ID + OPTION-FRAME-ID
OPTION-LIST	=	OPTION-LIST-HEADER + {OPTION-LIST-LINE}
OPTION-LIST-HEADER	=	OPTION-STEP-ID
OPTION-LIST-LINE	=	OPTION-NBR + NEXT-COURSE-STEP
OPTION-NBR	=	Identifier of option choice for a specific Option Step
OPTION-STEP	=	OPTION-STEP-ID + COURSE-REF
OPTION-STEP-COMPLETION	=	OPTION-STEP-ID + STUDENT-ID + NEXT-STEP-ID
OPTION-STEP-CONTENT-RPT-LINE	=	OPTION-STEP-ID + OPTION-FRAME-ID
OPTION-STEP-CONTENTS	=	See Entity Relation Diagram
OPTION-STEP-DEV-ACTIVITY-DATA	=	OPTION-STEP-TO-ADD | OPTION-STEP-TO-MODIFY | OPTION-STEP-TO-DISPLAY | COURSE-OPTION-STEP-LIST-REQ | OPTION-STEP-TO-DELETE | ENTIRE-OPTION-STEP-TO-DELETE
OPTION-STEP-ID	=	COURSE-STEP-ID where STEP-TYPE = "OPTION"
OPTION-STEP-LIST	=	OPTION-STEP-LIST-HEADER + {OPTION-STEP-LIST-LINE}
OPTION-STEP-LIST-HEADER	=	COURSE-ID + DATE + TIME
OPTION-STEP-LIST-LINE	=	OPTION-STEP-ID + OPTION-FRAME-ID
OPTION-STEP-PARAMETER-SCREEN	=	OPTION-STEP-ID + OPTION-FRAME-ID
OPTION-STEP-RQMNT		COURSE-ID + OPTION-STEP-ID + STUDENT-ID
OPTION-STEP-TO-ADD	=	COURSE-ID + OPTION-STEP-ID
OPTION-STEP-TO-DELETE	=	COURSE-ID + OPTION-STEP-ID
OPTION-STEP-TO-DISPLAY	=	COURSE-ID + OPTION-STEP-ID
OPTION-STEP-TO-MODIFY	=	COURSE-ID + OPTION-STEP-ID
PERFORMANCE-HEADER-LINE	=	STUDENT-ID
PERFORMANCE-QUEUE	=	{QUESTION-SUCCESS-FLAG}

PERFORMANCE-RQMNTS	=	QUESTION-GROUP-ID + MAX-QUEST-ASKED + MIN-QUEST-ASKED + NBR-CORRECT-ANSWERS-REQUIRED + NBR-LAST-QUEST
QUEST-DEV-ACTIVITY-DATA	=	QUESTION-TO-ADD | QUESTION-TO-ADD-TO-GROUP | QUESTION-TO-MODIFY | QUESTION-TO-DISPLAY | QUESTION-GROUP-TO-LIST | QUESTION-TO-DELETE
QUEST-GROUP-DEV-ACTIVITY-DATA	=	QUESTION-GROUP-TO-ADD | QUESTION-TO-ASSIGN-GROUP | QUESTION-GROUP-TO-ASSIGN-TO-STEP | QUESTION-GROUP-LISTING-REQ | QUESTION-GROUP-TO-DELETE | ENTIRE-QUESTION-GROUP-TO-DELETE | QUESTION-TO-REMOVE-FROM-GROUP
QUEST-STEP-DEV-ACTIVITY-DATA	=	QUESTION-STEP-TO-ADD | QUESTIONS-STEP-TO-MODIFY | QUESTION-STEP-TO-DISPLAY | COURSE-QUESTION-STEPS-TO-LIST | QUESTION-STEP-TO-DELETE | ENTIRE-QUESTION-STEP-TO-DELETE
QUESTION	=	QUESTION-ID + GRAPHICS
QUESTION-CONTENT-RPT-LINE	=	QUESTION-ID + DEFINED-FLAG + ANSWER-COUNT
QUESTION-DEVELOP-SCREEN	=	QUESTION-ID + GRAPHICS
QUESTION-DISPLAY	=	{QUESTION-GRAPHICS} + {STUDENT-ANSWER-DISPLAY} + {CORRECT-ANSWER-DISPLAY}
QUESTION-DISPLAY-MAP	=	pixel status for a question display
QUESTION-GRAPHICS	=	Display of GRAPHICS for QUESTION
QUESTION-GROUP	=	QUESTION-GROUP-ID
QUESTION-GROUP-CONTENT-RPT-LINE	=	QUESTION-GROUP-ID + {QUESTION-ID}
QUESTION-GROUP-ID	=	Unique identifier for Group of Questions
QUESTION-GROUP-LISTING	=	QUESTION-GROUP-LISTING-HEADER + {QUESTION-GROUP-LISTING-LINE}

QUESTION-GROUP-LISTING-HEADER	=	DATE + TIME + COURSE-ID
QUESTION-GROUP-LISTING-LINE	=	QUESTION-GROUP-ID + QUESTION-STEP-ID
QUESTION-GROUP-LISTING-REQ	=	Trigger
QUESTION-GROUP-TO-ASSIGN-TO-STEP	=	COURSE-ID + QUESTION-STEP-ID + QUESTION-GROUP-ID
QUESTION-GROUP-TO-ADD	=	QUESTION-GROUP-ID
QUESTION-GROUP-TO-DELETE	=	QUESTION-GROUP-ID + COURSE-ID
QUESTION-GROUP-TO-LIST	=	QUESTION-GROUP-ID
QUESTION-ID	=	Unique identifier for question
QUESTION-IN-PROGRESS	=	QUESTION-ID
QUESTION-INSTANCE	=	QUESTION-REF + STUDENT-REF + DATE + TIME
QUESTION-LISTING	=	QUESTION-LISTING-HEADER + {QUESTION-LIST-LINE}
QUESTION-LIST-LINE	=	QUESTION-ID
QUESTION-LISTING-HEADER	=	QUESTION-GROUP-ID + DATE + TIME
QUESTION-RQMNT	=	QUESTION-GROUP-ID + STUDENT-ID
QUESTION-SEQ-RPT-LINE	=	QUESTION-STEP-ID + DETACHED-FLAG + SUCCESS-COURSE-STEP + FAILURE-COURSE-STEP
QUESTION-STEP	=	QUESTION-STEP-ID + COURSE-REF + MAX-NBR-TRIES + MAX-QUEST-ASKED + MIN-QUEST-ASKED + NBR-CORRECT-ANSWERS-REQUIRED + NBR-LAST-QUEST
QUESTION-STEP-COMPLETION	=	QUESTION-STEP-ID + STEP-COMPLETION-STATUS + STUDENT-ID

QUESTION-STEP-CONTENT-RPT-LINE	=	QUESTION-STEP-ID + PERFORMANCE-RQMNTS
QUESTION-STEP-CONTENTS	=	See Entity Relation Diagram
QUESTION-STEP-ID	=	COURSE-STEP-ID where STEP-TYPE = "QUESTION"
QUESTION-STEP-LISTING	=	QUESTION-STEP-LISTING-HEADER + {QUESTION-STEP-LISTING-LINE}
QUESTION-STEP-LISTING-HEADER	=	DATE + TIME + COURSE-ID
QUESTION-STEP-LISTING-LINE	=	QUESTION-STEP-ID + QUESTION-GROUP-ID
QUESTION-STEP-PERFORMANCE	=	NBR-QUEST-ANSWERED + NBR-CORRECT-ANSWERS + PERFORMANCE-QUEUE
QUESTION-STEP-PARAMETER-SCREEN	=	COURSE-ID + QUESTION-STEP-ID + QUESTION-STEP-PARAMETERS + SUCCESS-COURSE-STEP + FAILURE-COURSE-STEP
QUESTION-STEP-PARAMETERS	=	NBR-CORRECT-ANSWER-REQUIRED + MIN-QUEST-ASKED + MAX-QUEST-ASKED + NBR-LAST-QUEST + QUESTION-GROUP-ID + MAX-NBR-TRIES
QUESTION-STEP-RQMNT	=	QUESTION-STEP-ID
QUESTION-STEP-TO-ADD	=	COURSE-ID + QUESTION-STEP-ID
QUESTION-STEP-TO-DELETE	=	COURSE-ID + QUESTION-STEP-ID
QUESTION-STEP-TO-DISPLAY	=	COURSE-ID + QUESTION-STEP-ID
QUESTION-STEP-TO-MODIFY	=	COURSE-ID + QUESTION-STEP-ID
QUESTION-SUCCESS-FLAG	=	"SUCCESS" | "FAILURE"
QUESTION-TO-ADD	=	QUESTION-ID
QUESTION-TO-ADD-TO-GROUP	=	QUESTION-GROUP-ID + QUESTION-ID
QUESTION-TO-ASSIGN-GROUP	=	QUESTION-GROUP-ID + QUESTION-ID
QUESTION-TO-DELETE	=	QUESTION-ID + COURSE-ID
QUESTION-TO-DISPLAY	=	QUESTION-ID
QUESTION-TO-MODIFY	=	QUESTION-ID

QUESTION-TO-REMOVE-FROM-GROUP	=	QUESTION-GROUP-ID + QUESTION-ID
REPORT-DEV-ACTIVITY-DATA	=	VALIDATION-RPT-REQ | CONTENT-RPT-REQ | SEQUENCE-RPT-REQ
RIGHT-WRONG-SW	=	"RIGHT" | "WRONG"
SCREEN-MSG	=	Text to be displayed on a dedicated portion of the screen
SCREENS	=	{SCREEN-MSG} + {QUESTION-STEP-PARAMETER-SCREEN} + {QUESTION-DEVELOP-SCREEN} + {DISPLAY-QUESTION-SCREEN} + {ANSWER-AREA-SCREEN} + {LESSON-STEP-PARAMETER-SCREEN} + {LESSON-FRAME-DEVELOP-SCREEN} + {OPTION-STEP-PARAMETER-SCREEN} + {OPTION-FRAME-DEVELOP-SCREEN} + {START-STEP-DISPLAY-LINE} + {GRAPHIC-DISPLAY}
SEQ-RPT-HEADER	=	COURSE-ID + DATE + TIME
SEQUENCE-RPT	=	SEQ-RPT-HEADER + {QUESTION-SEQ-RPT-LINE} + {LESSON-SEQ-RPT-LINE} + {OPTION-SEQ-RPT-LINE + {OPTION-CHOICE-SEQ-RPT-LINE}}
SEQUENCE-RPT-REQ	=	COURSE-ID
START-STEP-DEV-ACTIVITY-DATA	=	STARTING-STEP-FOR-COURSE | STARTING-STEP-DISPLAY-REQ
START-STEP-DISPLAY-LINE	=	COURSE-ID + COURSE-STEP-ID
STARTING-STEP-DISPLAY-REQ	=	COURSE-ID
STARTING-STEP-FOR-COURSE	=	COURSE-ID + COURSE-STEP-ID
STARTUP	=	STUDENT-ID
STEP-COMPLETION-STATUS	=	"SUCCESS" | "FAILURE"
STEP-TYPE	=	"LESSON" | "QUESTION" | "OPTION"
STUDENT	=	STUDENT-ID + STUDENT-NAME
STUDENT-ACCESS-TO-ADD	=	STUDENT-ID + COURSE-ID
STUDENT-ACCESS-TO-REMOVE	=	STUDENT-ID + COURSE-ID

STUDENT-ANSWER-DISPLAY	=	Display of ANSWER-COORDINATES chosen by student
STUDENT-NAME	=	first and last name of student
STUDENT-ID	=	Unique identifier for a student
STUDENT-INPUT	=	{ANSWER-COORDINATES} + {OPTION-COORDINATES}
STUDENT-LIST	=	STUDENT-LIST-HEADER + {STUDENT-LIST-LINE}
STUDENT-LIST-HEADER	=	DATE + TIME
STUDENT-LIST-LINE	=	STUDENT-ID + STUDENT-NAME
STUDENT-PERFORM-INQ	=	STUDENT-ID
STUDENT-PERFORMANCE	=	See Entity Relation Diagram
STUDENT-PERFORMANCE-RPT	=	PERFORMANCE-HEADER-LINE + {COMPLETE-COURSE-LINE} + {INPROGRESS-COURSE-LINE} + {NOT-STARTED-COURSE-LINE}
STUDENT-TO-ADD	=	STUDENT-ID + STUDENT-NAME
STUDENT-TO-REMOVE	=	STUDENT-ID + STUDENT-NAME
SUCCESS-COURSE-STEP	=	COURSE-STEP-ID
XCOORDINATE	=	Horizontal coordinate on screen
YCOORDINATE	=	Verticle coordinate on screen
YES-NO-RESPONSE	=	Response of yes or no to prompt

ENTITY-RELATION DIAGRAM

The box represents a object type. The diamond represents a relationship identifying an association between object types. Instances of object types can be identified uniquely, play a role in the system, and can be described by data elements.

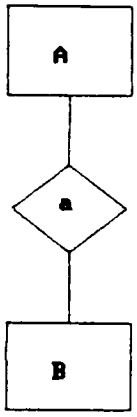


Fig 1: Object type A is associated to oject type B through relation a.

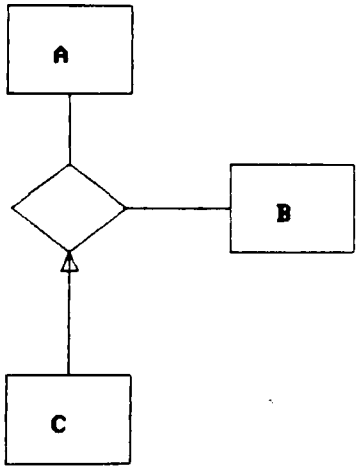


Fig 2: Object type C is an associative object type which functions both as an oject type (as data storage) and a relationship (it depends on the objects its connected to - A & B - for its existence).

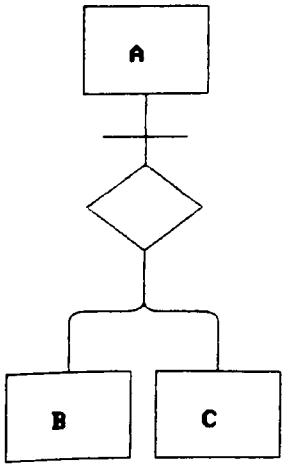


Fig 3 represents subtypes B & C and supertype A object types. A is described by data elements which apply to all subtypes. B & C are described by data elements not common to both.

APPENDIX E

OPERATING ENVIRONMENT

Hardware requirements:

IBM PC compatable operating under MS-DOS rev 3.1
Hercules Monochrome Adapter
or
IBM Graphics Adapter
Microsoft Mouse
Hard disk

Software requirements:

The following files are required:

AUTHOR.COM
AUTHOR.xxx where xxx is 3 digits (overlay files)
STUDENT.COM
STUDENT.xxx where xxx is 3 digits (overlay files)
MAKEDB.COM
MAKEFRX.COM
ERROR.MSG
14X9.FON (for Hercules graphics adapter)
8X8.FON (for IBM graphics adapter)
8X9.FON (for Zenith graphics adapter)
Microsoft PAINTBRUSH (optional)

The following files are created, updated and/or used temporarily by the Graphical Authoring System and should not be created, modified or deleted through another means:

*.FRM
*.FRX
*.ANS
*.HLP
COMPFILE.NDX
STEPFILE.NDX
ANOPFILE.NDX
PERFFILE.NDX
STEPFILE.DAT
ANOPFILE.DAT

Set Up requirements:

CONFIG.SYS- when the system is booted, parameters are identified in this file.
AUTOEXEC.BAT- executes graphics.exe which initializes the mouse.

APPENDIX F

REFERENCES

ASHT86 Ashton Interactive Training Corp., InterAct marketing brochure and information, 1986.

BORK81 Bork, Alfred, Learning with Computers, Digital Press, Bedford, Mass, 1981.

BYTE86 BYTE Magazine, 1985-86, Software and hardware reviews).

CHAR82 Chartrand, Marilyn J. and Constance Williams, Educational Software Directory - A Subject Guide to Microcomputer Software, Corporate Monitor Inc., 1982.

COMP86 Computer Teaching Corp., TenCore marketing brochure and information, 1986.

CREA00 Creative Computing, Ahl Computing Ins., Los Angeles, CA. (contains software reviews).

DIGI83 Digital Equipment Corp., Introduction to Computer-based Education, Digital Equipment Corp., 1983.

EPIE81 EPIE, EPIE Materials Report - Microcomputer Courseware/Microprocessor Games, EPIE Institute, Stonybrook, NY, 1981.

EPIE83 EPIE, EPIE & Consumer Union Micro-Courseware PRO/FILE and Evaluation, EPIE Institute, Stonybrook, NY, 1983.

FOLE82 Foley, James D. and Andries Van Dam, Fundamemtals of Interactive Computer Graphics, Addison-Wesley Publishing Company, Reading, Ma., 1982.

FREN80 Frenzel, Lou, The Personal Computer - Last Chance for CAI, Byte, pp 86-94, July 1980.

GRAD83 Grady, M. Tim and Jane D. Gawronski, Computers in Curriculum and Instruction, Association of Supervision and Curriculum Development, 1983.

GRAH85 Graham, Stephen N., Tools for Creating Lessons on the Computer, Computers in Life Science Education, pp 1 - 5, Jan 1985.

HEAR85 Heard, John Jr. and Barbara K. Heard,Video-Based Instruction as an enhancement to Computer Learning, Computers in Life Science Education, pp 43-45, June 1985.

HOYE73 Hoye, Robert E. and Anastasia C. Wang, Index to Computer Based Learning, Educational Technology Publications, Englewood Cliffs, New Jersey, 1973.

INST84 Instructional Computing Conference, Rochester Institute of Technology, June 1-3, 1984.

JONE85 Jones, M.E. and J.W. Mann and M. Aucone, Factors Influencing the Success of CAI Programs, Computers in Life Science Education, pp 61-63, Aug 85.

KOZM78 Kozma, Robert B., and Lawrence W. Belle and George W. Williams, Instructional Techniques in Higher Learning, Educational Technology Publications, Englewood Cliffs, New Jersey, 1978.

LOCA85 Locatis, Craig and Victor Carr, Factors to Consider when Choosing an Authoring System, Computers Life Science Education, pp 5 - 7, Jan 1985.

LATA82 Lathrop, Ann, Classroom Test Your Courseware Before You Buy It, Educational Computer, p. 34, Sept-Oct 1982.

LATB82 Lathrop, Ann, The Terrible Ten in Educational Programming, Educational Computer, p. 34, Sept-Oct 1982.

LEKA71 Lekan, Helen H., Index to Computer Assisted Instruction, Harcourt Brace Jovanovich Inc., New York, NY, 1971.

ONEI81 O'Neil, Harold F.,Jr., Computer-Based Instruction, A State-of-the-Art Assessment, Academic Press, New York, NY, 1981.

PCMA85 PC Magazine, 1985-86 (Software and Hardware reviews).

PLUM84 Plummer, Charles M., Faculty and Program Development Project Director, Computer Literacy Education & Training System, 1984.

RAPT86 Raptor Systems, THE AUTHOR and THE AUTHOR PLUS marketing brochure, 1986.

ROCK81 Rockart, John Fralick and Michael S. Scott Morton, Computers and Learning Process in Higher Education, McGraw-Hill Book Co., New York, NY, 1981.

SKIB85 Skiba, Diane J., Evaluation of Computer-Assisted Instruction Courseware, Computers in Life Science Education, pp 11 - 14, Feb 1985.

VANA84 VanAlstyne, Daryl, BOCES - Spencerport, 1984.